Computer system with touchpad support in operating system

Patent number: CN1139235

Publication date: 1997-01-01

Inventor: BERTRAM RANDAL LEE (US); COMBS JAMES LEE (US)

Applicant: IBM (US)

Classification:

- international: ***G06F3/041; G06F3/033; G06F3/038; G06F3/048;*** **G06F3/041; G06F3/033; G06F3/048;** (IPC1-7): G06F3/02; G06F13/00

- european: G06F3/048A3T

Application number: CN19951002040 19950217

Priority number(s): US19940210610 19940318

Also published as:

EP0675426 (A1)
US5875311 (A1)
US5613137 (A1)
JP7261923 (A)
EP0675426 (B1)
CN1133916C (C)
CA2142798 (C)

less <<

Abstract not available for CN1139235
Abstract of correspondent: **US5875311**

A computer system comprising a central processing unit (CPU) configured to accept coordinate type data from a touchpad or the like. The CPU has an operating system executing thereon with special support for interfacing to the touchpad. The operating system has the following capabilities: (1) mapping out geometric regions of the touchpad and assign the regions to specific region identifiers responsive to application programs and (2) determining the region identifier of a touched region and passing that region identifier to the application program. Support is also provided for changing the units of the commands used to define the regions.

[19]中华人民共和国专利局

[51]Int.Cl[6]
G06F 3/02
G06F 13/00

[12] 发明专利申请公开说明书

[21] 申请号 95102040.4

[43]公开日 1997年1月1日

[11] 公开号 CN 1139235A

[22]申请日 95.2.17

[30]优先权

[32]94.3.18 [33]US[31]210,610

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 R·L·伯特兰姆

J·L·康斯

[74]专利代理机构 中国专利代理(香港)有限公司

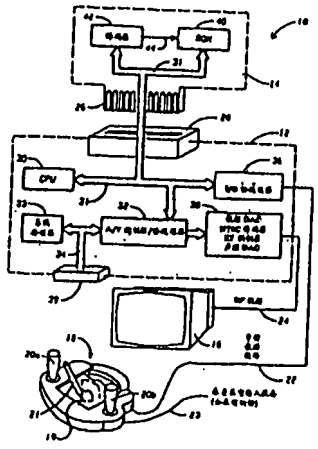
代理人 杜有文 马铁良

权利要求书 4 页 说明书 37 页 附图页数 11 页

[54]发明名称 在操作系统中支持触摸板的计算机系统

[57]摘要

一个包含一个中央处理单元（CPU）的计算机系统，配置为接受来自一个触摸板或类似设备的坐标型数据。CPU有一个在上面运行的操作系统，后者具有对触摸板的接口的特殊支持。该操作系统有下列能力：（1）映射出触摸板的几何区域，并赋予区域特定的区域标识符作为对应用程序的响应；（2）确定一个被触区域的区域标识符并把该区域标识符传送给应用程序。还提供对用来定义区域的命令的单位的改变的支持。



权 利 要 求 书

1. 一个计算机系统包含:

(a) 一个中央处理单元(CPU);

(b) 与所述 CPU 电路连通的一个存储器;

(c) 与所述 CPU 电路连通的一个外围接口电路, 其特征在于能够把来自至少一个具有接触敏感的表面的外部触摸板的坐标型数据送到所述 CPU; 以及

(d) 与所述 CPU 和所述外围接口单元相关联的触摸区域定义逻辑, 后者配置为定义触摸板的一个区域并响应来自在所述 CPU 上执行的至少一个应用程序的输入而把该区域与一个区域标识符相关联。

2. 权利要求 1 的计算机系统, 其特征在于还包含与所述 CPU 关联的区域识别逻辑, 后者配置为响应对触摸板表面的接触而确定坐标型设备的一个被触区域的区域标识符。

3. 权利要求 1 的计算机系统, 其特征在于所述区域定义逻辑的特征还在于能够定义触摸板表面的至少一个无效区域, 该区域的特征在于对至少一个无效区域的接触不被传送给至少一个应用程序。

4. 权利要求 1 的计算机系统, 其特征在于还包含触摸板映射单位逻辑, 后者配置为响应来自至少一个应用程序的输入而改变对所述区域定义例程的输入的单位。

5. 权利要求 1 的计算机系统, 其特征在于所述触摸区域定义逻辑包含与所述 CPU 电路连通的电路。

6. 权利要求 1 的计算机系统，其特征在于所述触摸区域定义逻辑包含在所述 CPU 上执行的可执行代码。

7. 权利要求 1 的计算机系统，其特征在于所述触摸区域定义逻辑配置为允许定义具有各种几何形状的区域。

8. 权利要求 1 的计算机系统，其特征在于所述触摸区域定义逻辑配置为允许定义三角形，矩形，和圆形区域。

9. 一个计算机系统包含

(a) 一个中央处理单元(CPU)；

(b) 一个与所述 CPU 电路连通的存储器电路；

(c) 与所述 CPU 和所述存储器电路连通的一个视频电路，用于产生一个相应于要在一个视频显示设备上显示的一个视觉图象的电信号；

(d) 与所述 CPU 电路连通的一个外围接口电路，用来把来自外部设备的信号送到所述 CPU；

(e) 一块暴露的板面，以允许用手指、铁笔或类似物体接触之；

(f) 一个用于可移去地把一个模板覆盖固定紧贴在所述板面上的夹持器；

(g) 一个紧贴所述板面的坐标传感器，用于产生至少一个电信号，配置为使所述电信号的累积对应于铁笔、手指或类似物体对所述板面或对紧贴所述板面的模板覆盖的接触的位置的坐标；

(h) 一个紧贴所述板面的刚性的基座；

(i) 与所述坐标传感器电路连通的坐标确定电路，用于确定铁笔、手指或类似物体对所述板面或对紧贴所述板面的模板覆盖的接触的位置的坐标；

(j) 与所述坐标确定电路和所述外围接口电路电路连通的接口电路，用于向它传送所确定的坐标；以及

(k) 与所述 CPU 和所述外围接口单元相关联的接触区域定义逻辑，配置为定义触摸板的一个区域并把该区域与一个区域标识符相关联，作为对来自在所述 CPU 上执行的至少一个应用程序的输入的响应。

10. 权利要求 9 的计算机系统，其特征在于还包含与所述 CPU 相关联的区域识别逻辑，后者配置为响应对触摸板表面的接触而确定该坐标型设备的一个触摸区域的区域标识符。

11. 权利要求 9 的计算机系统，其特征在于所述区域定义逻辑的特征还在于能够定义触摸板表面的至少一个无效区域，该区域的特征在于对至少一个无效区域的接触不被传送给至少一个应用程序。

12. 权利要求 9 的计算机系统，其特征在于还包含触摸板映射单位逻辑，后者配置为响应来自至少一个应用程序的输入而改变对所述区域定义例程的输入的单位。

13. 权利要求 9 的计算机系统，其特征在于所述触摸区域定义逻辑包含与所述 CPU 电路连通的电路。

14. 权利要求 9 的计算机系统，其特征在于所述触摸区域定义逻辑包含用于在所述 CPU 上执行的可执行代码。

15. 在一个具有一个中央处理单元 (CPU) 和一个与所述 CPU 电路连通并用于把一个不透明的具有一个触摸敏感表面的坐标型输入设备电连接到所述 CPU 的外围接口电路的计算机系统中，一个操作系统包含：

(a) 一个区域定义例程, 配置为定义坐标型设备的至少一个区域并响应在 CPU 上执行的至少一个应用程序的输入而把该区域与一个区域标识符相关联;

(b) 一个触摸板接口例程, 配置为从外围接口电路获得坐标型数据;

(c) 一个区域识别例程, 配置为响应对表面的接触而确定坐标型设备的一个区域的区域标识符; 以及

(d) 一个区域标识通信例程, 配置为把确定的区域标识符传送给至少一个应用程序。

16. 权利要求 15 的计算机系统, 其中

(a) 所述区域定义例程的特征还在于能够定义触摸板面的至少一个无效区域, 该无效区域的特征在于对至少一个无效区域的接触不被传送给至少一个应用程序;

(b) 所述区域标识通信例程的特征还在于不向至少一个应用程序传送对至少一个无效区域的接触。

17. 权利要求 15 的计算机系统, 其特征在于还包含一个触摸板映射单位例程, 它配置为响应从至少一个应用程序的输入改变对所述区域定义例程的输入的单位。

说　明　书

在操作系统中支持触摸板的
计算机系统

本发明总的来讲与和数字系统一起使用的坐标型指点设备有关，更具体地讲与一种在操作系统中具有对触摸板的支持的计算机系统有关。

视频图形计算机系统是众所周知的大受欢迎的消费产品。一个典型的系统包括一个数据处理单元，它与一台普通的电视机相连，用于显示一个游戏或其它应用程序的图象。数据处理单元从一块通常以盒式存储器的形式包装起来的只读存储器（RO M）接受控制软件。该卡可插入数据处理单元中，也可以拔出。至少有一种指点设备，例如鼠标器、游戏棒、触摸板、触摸屏、开关盘、或者光枪，也被连到数据处理单元，使用户可以输入供控制软件执行应用程序用的位置信息。

数据处理单元通常具有一个中央处理单元(CPU)和相联的易失性及非易失性的存储器(包括所有的随机存取存储器(RAM)和自举只读存储器(自举ROM))，一个电视(RF视频)信号发生器，和一个用来连接各种指点设备的输入/输出(I/O)处理器接口。这些设备是电路连通的。这些系统的一个与众不同的特点是使用一块母板或系统板把这些元件电联接在一起。

触摸板是坐标型指点设备，用于输入坐标型数据到计算机系统。

触摸板一般是一个压力敏感的有界平面，能够检测它表面上的局部的压力。当用户用手指、铁笔或类似物体接触这个表面时，触摸板确定所触的位置并把该位置的坐标通过产生的某种类型的信号报告给所接的计算机系统。作为回应，计算机执行与所触位置相关的功能，如果有的话。

一般地，触摸板的一个或多个区域被赋予系统内或应用程序内的某种功能，例如输入数据或提供方向输入。按照常规，用户从一个模板得知哪个功能与哪个区域相关联。模板是一张有图形设计的纸，一般贴在触摸板表面上。图形设计一般绘出或映射出触摸板表面的区域的轮廓而且这些区域一般加以标注以提示用户哪种功能与所映射出的区域中的哪一个相关联。

，在典型的触摸板系统中，应用程序从触摸板接受坐标位置数据信号。例如，当触摸板报告板面向上十五列向下十二行的地方被触，应用程序必须把坐标位置与映射到那个区域的功能联系起来。这就是说，应用程序必须对指示出哪一个图形图案中的区域被触的信号进行译码，进而决定哪种功能与所触位置相关联。

依赖于每个应用程序来决定哪个区域被触导致了许多问题。第一，应用程序变得依赖于硬件。一个特定的应用程序必须"知道"所使用的触摸板的分辨率(行和列的数目)，总体尺寸，以及数据格式，因而导致应用程序也许不能使用多种类型的触摸板来执行。随着技术的进步提高了触摸板的分辨率，现有程序可能没有灵活性以调节。第二，这需要应用程序员在应用程序中写进位置译码的代码。这样，每个应用程序必须有自己的区域确定例程，于是导致了不必要的重复劳动和可能的不一致性以及甚至支持触摸板区域的方式中的错误。

根据本发明，提供一个在操作系统中对触摸板的支持的计算机系统。该操作系统有如下能力：(1) 映射出触摸板的几何区域并且给这些区域赋予对应于应用程序的专有的区域标识符；(2) 确定一个触摸板区域的区域标识符并把该区域标识符传给应用程序。

这些普通功能给与一个触摸板接口的任务提供了一致性和灵活性。

本发明的这些及其它优点将随着对本发明的详尽描述而变得更为明显。

在溶进并且组成了本说明的一部分的附图中，示出了本发明的实施例，这些与上文对本发明的一般描述，以及下文的详细描述，共同用于例述本发明的原理。

图 1A 和 1B 是显示本发明的系统总体布局的方框图；

图 1C 是显示用于本发明的系统中的视频数字到模拟转换器的细节的示意图；

图 2A 是本发明的输入设备的顶视平面图，示出了有缺省模板的触摸板，两个游戏棒，和一个模板覆盖；

图 2B 是本发明的输入设备的顶视平面图，其中一个模板覆盖被插入了模板覆盖夹持器中；

图 2C 是图 2B 所示的本发明的输入设备的一部分的一个放大的片段的平面图；

图 2D 是沿由图 2C 中的线 2D－2D 所标明的平面所截的一个剖面图；

图 2E 是沿由图 2C 中的线 2E－2E 所标明的平面所截的一个剖面图；

图 2F 是沿由图 2A 中的线 2F－2F 所标明的平面所截的一个剖面图;

图 2G 是图 2A 所示的本发明的输入设备的一个片段顶部平面图, 其中为清晰起见而部分省略了;

图 2H－2K 是一个显示识别模式的模板覆盖的一个边缘的几个不同实施例的底部平面图;

图 2L 是本发明的输入设备的一个前部正视图;

图 2M 是沿由图 2A 中的线 2M－2M 所标明的平面所截的部分剖面图, 它显示本发明的输入设备所用的游戏棒的细节; 以及

图 3 是显示本发明的输入设备的电路的方框图。

图 4 显示了以本发明的支持触摸板的操作系统设计的一个触摸板的一个可能的配置图。

参考图 1A 和 1B, 显示了一个本发明的计算机系统 10。据图 1A 所示, 系统 10 包括一个数据处理单元 12, 有一个程序盒式存储器以可移去方式与之相连。同样连到数据处理单元 12 上的还有一台标准电视机 (TV) 16 和一个输入设备 18, 后者有一个触摸板 19 和两个游戏棒 20a, 20b。输入设备 18 把由手指、铁笔 21 或者类似的东西在触摸板上所触的位置所对应的坐标型数据送至数据处理单元 12。此外, 输入设备 18 还向数据处理单元 12 发送由游戏棒 20a 和 20b 的运动所对应的方向型数据。虽然图 1A 中没有示出, 但标准电视机 16 可由一对扬声器和一个接受复合视频信号的显示设备来代替。输入设备 18 通过一条串行数据链路 22 与数据处理单元 12 相联。电视机 16 通过一条RF 视频线 24 与数据处理单元 12 相联。

盒式存储器 14 有一个边缘卡片连接器, 后者用 26 指出, 并与盒

式存储器连接器 28 连接，因而把盒式存储器 14 中的设备与数据处理单元 12 中的设备进行电连通。

数据处理单元 12 由一个中央处理单元(CPU)30(它有一条与之相连的系统总线 31)，一个声频/视频(A/V)控制器/协处理器 32，一个系统存储器 33（它连到由 A/V 控制器/协处理器 32 从系统总线 31 产生的副系统总线 34 上)，第一和第二译码器芯片 (未示出)，一个 I/O 协处理器 36，两个盒式存储器连接器 (一个由 28 指出，另一个未示出)，需要用来产生声频和视频信号的附加电路 38，以及一个扩展连接器 39 组成。这些设备是如图中数字所示以电路连通方式连接起来。附加电路 38 由图 1B 示出并将在伴随图 1B 的文字中做更为详尽的讨论。

CPU 30 产生多条总线：一个数据总线，一个地址总线，和一条控制总线，这是本技术中众所周知的。这三个总线的集合称为系统总线 31。在较佳实施例中，CPU 30 是一块 80376，由 Intel 公司 (3065 Bowers Ave., Santa Clara, California, 95051) 生产。80376 是著名的 80386SX 的一个变种，后者在本技术中众所周知并同样可从 Intel 公司得到。80376 与 80386SX 的不同之处在于 80376 以 32 位模式启动，而不是 16 位模式。具体地说，CR0 寄存器被强置为 0011H (十六进制的 0011) 状态，其中位 0 被强置为逻辑 1，实际上使 376 以 32 位模式工作。分页被启动以允许虚拟 386 操作。

A/V 控制器/协处理器 32 从系统总线 31 产生三条附加的通用 I/O 译码线 (GPIO1, GPIO2, GPIO3)，每条线提供一个 32 位 I/O 地址范围。可以使用通用译码器来提供低电平有效使能信号到 A/V 控制器/协处理器 32 之外的设备中。在数据处理单元 12 里，使用通用

译码器来把地址范围译码到I/O协处理器36 (GPIO1)，和两个盒式存储器连接器 (GPIO2和GPIO3)。下文将讨论A/V控制器/协处理器的其余电路。

系统存储器33由屏幕RAM，系统RAM和自举ROM组成(全都未示出)。板上的屏幕RAM和系统RAM是一兆字节的32位DRAM。合适的DRAM是用一对由东芝生产的256千字节乘以16位的TCS1470BJ存储器芯片配置成32位存储器。CPU 30的地址空间的一部分被译码到A/V控制器/协处理器32中的多个8位寄存器。所有的内部位置都是在偶数地址边界上；可以在适当的地方进行字宽I/O读写。在此特定实施例中，不允许在字宽寄存器上执行字节宽的写，而且不能使用I/O周期来访问奇数地址。

自举ROM总是16位宽。自举ROM由两块27C512可擦除可编程只读存储器组成，它可由许多厂家生产，因而共有128K自举ROM。紧跟一个复位，从F20000H到FFFFFFH的一个一兆字节窗口在16兆字节地址范围中被重复，该窗口包含了ROM和内部存储器。

系统存储器33由多个设备共享。A/V控制器/协处理器32是系统存储器33的仲裁器；因此，系统总线31被A/V控制器/协处理器32修改成一个副系统总线34(包括一个副数据总线，一个副地址总线和一个副控制总线，全部未示出)。这样，系统存储器33是通过副系统总线34被访问的。

I/O协处理器36把CPU 30连接到多个输入设备，例如输入设备18和可选的设备如键盘(未示出)，控制器(未示出)，鼠标器(未示出)，以及打印机(未示出)。在该较佳实施例中，I/O协处理器36是

由摩托罗拉公司生产的预编程的 MC68HC705C8 （此后称为"68HC705"），它运行在 2MHZ。68HC705 I/O 协处理器 36 通过把 68HC705 配置成一个外部设备来与 CPU 30 连接：(1) PA0－PA7 接到数据总线的 D0－D7 (2) PB7、PB1 和 PB2 分别接到控制和地址总线的 GPIO1 (由 A/V 控制器/协处理器 3 2 译码的一个 32 位地址范围)、A1 和 A2；(3) PB3，PB4 和 PB5 分别接到控制总线的 ADS, READY, 和 W/R。I/O 协处理器 36 被 A/V 控制器/协处理器译码为具有 I/O 空间中的四个 16 位地址（此处指称为 AS0, AS2，AS4 和 AS6)。

68HC705 中的程序以如下方式连接到 CPU 30。68HC705 设计为直接挂到处理器总线上，并表现为 CPU 30 的一个 I/O 接口。一对内部锁存器保持在各个处理器之间传送的数据直到另一个准备好接收。每个处理器的状态位指示出数据锁存器的状态。每个处理器都能通过检查状态位以得知前次数据是否已被读出以及是否有新数据待读。

I/O 协处理器 36 尤其实现下述功能：(1) 一个 50 毫秒定时器，(2) 一个从输入设备接收通讯包的串行控制器链路，(3) 一个盒式存储器/扩展传感器，用于确定在每一个盒式存储器连接器中是否有一个盒式存储器 14，以及在扩展连接器中是否有一个扩展设备或者 CD 驱动器，(4) 一个系统复位，和 (5) 一个 I^2C 非易失性 RAM (NVRAM) 接口。I/O 协处理器 36 还实现了一个可选的 DSA 密致盘控制串行链路以允许与可选的 CD 设备的通信。

50 毫秒定时器是通过把 68HC705 I/O 协处理器 36 的监视定时器配置成每到 50 毫秒间隔就到时来实现的。每次监视定时器到时，I

/O 协处理器 36 就使用 A/V 控制器/协处理器 32 的模拟中断 U 来中断 CPU 30 (响应 I/O 协处理把 AI0 拉至低电平，A/V 控制器/协处理器通过 IRQ 线来中断 CPU)。CPU 通过把字节 0F0H 或者字节 00H 写入 I/O 端口 AS0 来分别启动及禁止该 50 毫秒定时器。该定时器缺省状态是启动。

在 CPU 的中断确认周期中，A/V 控制器/协处理器确定中断服务例程的地址。中断服务例程使 CPU 30 从相应于 I/O 协处理器的 16 位 I/O 端口 AS0 读取一个或多个字节。在每一次读取 I/O 端口 AS0 时，A/V 控制器/协处理器 32 选择 I/O 协处理器 36，因而允许数据在 CPU 30 和 I/O 协处理器之间传输。

，响应 50 毫秒中断，I/O 协处理器 36 总会有一个字节传输到 CPU。该字节的低半字节包含了自上次中断确认周期以来的 50 毫秒到时的次数；高半字节包含要传送到 CPU 的I/O 设备消息的数目。如果 50 毫秒定时器被禁止，该字节的低半字节将为 0。如果收到的消息已超过 15 个，那么 15 被送到高半字节而余下的消息等下次传输时再发送。根据此第一字节的内容，CPU 可能会从 I/O 协处理器 36 读取随后的字节，后者在多数情况下会是从输入设备来的数据包。一般地，输入设备仅会在其各自的状态改变时才会发消息，因而使消息传送频率非常低。

输入设备 18 和所有其它输入设备都通过串行数据链路 22 接到 I/O 协处理器 36。各自的输入设备 (比如输入设备 18) 把控制设备的运动转换成适于沿着串行链路 22 传输的格式。输入设备 18 把数据包沿着串行链路 22 发送到系统单元 12。正如下文所述，数据包的结构随输入设备的类型而不同。坐标型设备(鼠标器，模拟游戏棒，触摸

板等) 的数据包结构与开关闭合型设备 (键盘, 数字游戏棒, 开关盘等) 的不同。

串行控制器链路 22 包含三条线: 一条数据接收线, 一条 VCC (+5 伏直流电) 线, 和一条地线。68HC705 使用 68HC705 的 PD0/RDI 引脚来实现控制器串行链路的数据接收线。该引脚用众所周知的异步格式设计为与串行设备的接口。串行传输有如下格式: 每秒 4800 位, 无校验, 8 个数据位和一个停止位。一种时钟同步格式可用作替代。串行控制器链路 22 通过本技术中众所周知的一个六导线小引脚插头连接器 (six－conductor mini－din plug connector) (未示出) 接到外部设备。输入设备是菊花链式的, 这样只有一个设备物理地接到数据处理单元 12。例如, 如果一个所谓鼠标指点设备被增加到系统 10 中, 鼠标被接到输入设备 18, 后者又接到处理单元 12。

盒式存储器传感与扩展传感是用来确定在每一个盒式存储器连接器中或扩展连接器中是否存在一个盒式存储器 14 的, 它是通过令 I/O 协处理器 36 查询盒式存储器连接器 28 的一个引脚来实现的。该脚被系统板上的一个适当的上拉电阻 (未示出) 拉至逻辑 1, 一个正确连接的盒式存储器把该脚拉至逻辑 0。这样, 每个盒式存储器传感的一个 1 表明不存在一个盒式存储器 14 而一个 0 则表明存在一个盒式存储器 14。类似地, 扩展传感的一个 1 表明不存在一个扩展设备, 例如一个可选的 CD 驱动器, 而一个 0 表明存在一个扩展设备。

复位是这样实现的: 赋予 I/O 协处理器对 A/V 控制器/协处理器的复位信号的控制, 后者接着控制了 CPU30 的复位信号。CPU 30 可以通过命令 I/O 协处理器 36 复位 A/V 控制器/协处理器, 接着后

者又复位 CPU 30 这样来复位系统 10。CPU 通过把字节 0FFH 写入I/O 端口 AS0 来使 I/O 协处理器产生一个系统复位。另外，I/O 协处理器 36 监视可选的系统复位开关 (未示出) 并且当它检测到一次开关闭合时便使系统复位。

最后，I/O 协处理器实现读、写以及检验 512 字节非易失性系统 RAM 的内容的一个 I^2C 非易失性 RAM (NVRAM) 接口。NVRAM (未示出) 由菲力普半导体公司生产的 PCF8594 构成，它通过 I^2C 接口与 I/O 协处理器电路连通。可以级联多个 PCF8594 来提供更多的 NVRAM 能力。为了访问 NVRAM，需要使用一个三字节序列。所有三个字节都通过 I/O 端口 AS0 来访问。由 CPU 写入 I/O 协处理器的第一字节表明传输是读还是写并且给I/O 协处理器一个段地址。这一字节的低半字节表明传输类型：01H 表明在 NVRAM 写入而 02H 表明从 NVRAM 读出。这一字节的高半字节是一个 4 位段号，对应于 NVRAM 的一个 256 字节段。对 512 字节的 NVRAM，只使用底端两个段(0 和 1)。下一字节对于读和写是相同的—下一字节是 CPU 写入，是在段内被访问的字节的地址。最后一个字节或者由 CPU 写入 I/O 协处理器或者由 CPU 从 I/O 协处理器读出，它是从 NVRAM 读出或向它写入的数据字节。

作为替代，I/O 协处理器可用其它方法实现。例如，一个三态可读移位寄存器可能适于从串行数据链路 22 接收信息。在这种情况下，CPU 30 周期性地读移位寄存器来访问从输入设备来的数据包。

第一译码芯片 (未示出) 与 CPU 30、A/V 控制器/协处理器 32 以及两个盒式存储器连接器 (另一个未示出) 电路连通。第一译码芯片接纳系统总线 31 的高端两条地址线作为输入，把 80376 CPU 30

的 16 兆字节地址空间译码成四个 4 兆字节区域，后者由三条芯片选择线代表：两条用于盒式存储器连接器 28（另一个未示出），一条用于 A/V 控制器/协处理器 32。高端四兆字节和低端四兆字节被译码到 A/V 控制器/协处理器芯片选择，其余两个四兆字节区域被译码到两个盒式存储器连接器芯片选择。

第二译码芯片（未示出）是用来实现扩展连接器 39 的芯片选择的。第二译码芯片沿副系统总线 34 与 A/V 控制器/协处理器和扩展连接器 39 电路连通。第二译码芯片允许 A/V 控制器/协处理器 32 来对从 F20000H 开始的 128K 的系统 ROM 块译码。从 F40000H 到 FFFFFFH 的范围由第二译码芯片译码，为扩展连接器 39 所用。由第二译码芯片译码的这一 ROM 块被用来通过扩展连接器 39 来为系统 10 增加 ROM。

数据处理单元 12 还有一对盒式存储器连接器（一个由 28 表示，另一个未示出）用来放置一个盒式存储器 14 与 CPU 30 和其它系统部件电路连通。盒式存储器 14 通过一个镀金 62 引脚（两排各 31 导线）边缘接插件 26 接到数据处理单元 12 的连接器 28 上。处理器单元 12 有两个盒式存储器连接器 28 来接纳边缘接插件 26 的边缘接插。盒式存储器 14 有镀金的边缘接插以对应于连接器 28 的导线，使盒式存储器 14 可插入地连接到处理器单元 12。下列信号是通过连接器 28（另一个未示出）与外部设备通信的：系统总线 31 信号，一条盒式存储器传感线，电源，地，模拟中断 1 或 2（每个盒式存储器有唯一的中断），GPIO2 或 3（每个盒式存储器有唯一的片选），一条锁定线（它是 80376 和80386SX 系统总线 31 的一个典型信号），以及由第一译码芯片产生的一个盒式存储器选择信号。作为替代，接到一个可

选的 CD 驱动器所需的信号也可以通过盒式存储器连接器 28 接到外部设备。

另外，处理器单元 12 还有一个 112 脚 (两排各 56 脚) 边缘卡扩展连接器 39。扩展连接器 39 允许为系统存储器 33 增加更多的存储器以及增加多种其它特点。连接到扩展连接器 39 上的设备有镀金的卡边缘以对应于扩展连接器，使得设备可插入地接到处理器单元 12 上。下列信号是通过扩展连接器 39 与外部设备通信的：副系统总线信号，一条扩展连接器 39 传感线，电源，地，CAS 和 RAS 线，一个扩展连接器 39 选择信号 (它是由第二译码芯片产生的)。

程序盒式存储器 14 由一个程序 ROM 40 和一个译码器 42 组成。作为替代，译码器 42 可以设计在处理器单元 12 之内。程序 ROM 40 包含只读存储器格式的适于在 CPU 30 上运行的代码。作为替代，其它类型的存储器，例如有后备电池的 RAM，也可以用作盒式存储器 14 中的存储设备。如图 1A 所示，程序 ROM 40 与 CPU 30 电路连通。

盒式存储器 14 内部的地址译码器 42 把全部宽度的地址总线译码到适用于程序 ROM 40 的存储器范围，并产生 ROM 40 所需的一个片选信号 44，这在本技术中是众所周知的。地址译码器 42 是用一块 16V8 可编程阵列逻辑 (PAL) 来实现的，后者在本技术是众所周知的，并由许多厂家生产，例如 AMD 公司。如果译码器 42 是设计在处理器单元内部，那么选择 44 就通过连接器 26 与 ROM 40 电路连通。

现在参考图 1B，显示出图 1A 中的附加电路 38 接到 A/V 控制器/协处理器 32。附加电路38 由 4 个设备组成、一个视频数－模转换器 (视频 DAC) 50，一个 NTSC/PAL ("PAL" 指的是著名的欧洲电视

信号标准）编码器 52，一个 RF 调制器 56，一个声频数－模转换器/模－数转换器/压缩器/解压缩器 5 4 (ADC/DAC/CODEC)。每一个都如图所示地连接。

声频/视频控制器/协处理器 (A/V 控制器/协处理器) 32 的电子电路大部分包含在一块大规模定制逻辑芯片中，称为一个 ASIC (应用专用集成电路)。一个满足此处描述的 A/V 控制器/协处理器 32 可以从 MSU 公司买到 (270 Upper 4th Street, Witan Gate West, Gentral Milton Keynes, MK9 IDP England)。A/V 控制器/协处理器 32 包含一个处理器接口 60，一个处理器高速缓存 62，一个内存接口/刷新 64，一个视频控制器66，一个中断控制器 6 8，一个视频阻击器 (blitter) 70，一个可选的 CD 块译码器，一个数字信号处理器 (DSP) 74，以及一个 DSP 存储器 76。处理器接口 60，内存接口/刷新64 和视频控制器 66 被合称为视频/存储器控制器 67。系统存储器 33，中央处理单元30，以及其它设备都位于 A/V 控制器/协处理器 32 的外面。

A/V 控制器/协处理器 32 从系统总线 31 产生副系统总线 34，因而把 CPU 30 与系统存储器 33 分隔开。这样，副系统总线 34 把各种设备电连接到系统存储器 33。副系统总线34 由六种可能的总线主控设备所共享 (依优先级从高到低的顺序，分别是)：存储器刷新 64，视频控制器 66，一个可选的 CD 块译码器 (未示出)，DSP 74，阻击器 (blitter) 70，以及 CPU 30 (通过处理器接口 60)。在同一时间只能有一个总线主控设备可以控制副系统总线 34。视频/存储器控制器内部的仲裁器控制此处所述设备的优先级的改变，并与 A/V 控制器/协处理器 32 内部的所有设备电路连通。例如，CPU 30 在所有总线主

控设备中的优先级最低，直到一个中断发生为止。这样，仲裁器既与CPU 接口60，也与中断控制器 68 电路连通。

在为 CPU 预取指令意义上高速缓存 62 并非是一个高速缓存。实际上，高速缓存 62 是位于 F14000H 到 F143FFH 的一块 512X16 位的静态 RAM，它被 CPU 用于变量，堆栈，或者程序代码以加速程序的执行。

视频/存储器控制器 67 (处理器接口 60，存储器接口/刷新 64，和视频控制器 66) 控制副系统总线 34，并为接到副系统总线 34 上的存储器设备提供存储器定时信号 (例如CAS，RAS，写使能等)，这在本技术中是众所周知的。它在视频线期间将总线主控设备操作挂起一小段时间以便取视频显示数据，以及刷新动态 RAM (DRAM)。它还控制与 CPU30 的接口。

视频控制器 66 有一个灵活的视频定时发生器，它可以被编程以适合于不同的电视标准和最高 640×480 的 VGA 标准的监视器。准确的视频格式是由设置 A/V 控制器/协处理器内部的不同的寄存器来控制的：水平周期，水平同步、水平消隐结束，水平消隐开始，水平显示开始，水平显示结束，水平取开始，水平取结束，水平垂直同步，垂直周期，垂直同步，垂直消隐结束，垂直消隐开始，垂直显示开始，垂直显示结束，视频中断，以及光笔寄存器。视频控制器 66 有三种可用的色彩分辨率：每象素 4 位，每像素 8 位，和每象素 16 位。屏幕的内存映象并不受限于视频显示宽度，而是独立定义。

视频/存储器控制器 67 把 16 兆字节的 80376 CPU 30 的地址范围译码到下述内存映象：1 兆字节系统存储器 RAM (000000H －0FFFFFH)，第一盒式存储器 ROM 的 4 兆字节

(400000 - 7FFFFFH)，第二盒式存储器 ROM 的 4 兆字节(800000H - BFFFFFH)，用于 A/V 控制器/协处理器的 64 千字节内部存储器 (F10000H - F1FFFFH)，以及一块 128 千字节的系统 ROM (FE0000H - FFFFFFH)。64 千字节的内部存储器包括调色板 RAM，阻击器 (blitter) 的寄存器，以及 DSP 的寄存器和存储器。调色板地址范围如前所述。阻击器 (Blitter) 的寄存器的范围是 F10400H - F107FFH。DSP 存储器的范围是 F10800H - F18000H。

如果可选的 CD 驱动器被增加到系统中，下述区域被增至内存映象：另外 1 兆字节的系统 RAM (100000H - 1FFFFFH) 和用于 CD 驱动器的 128 千字节 (FC0000H - FDFFFFH)。

中断控制器 68 把六种中断连到 CPU 30：视频中断 (最高优先级)，模拟中断 0 (AI0)，模拟中断 1 (AI1)，模拟中断 2 (AI2)，CD 块译码器中断，以及 DSP 中断 (最低优先级)。当 CPU 30 在执行中断应答周期时，中断控制器自动清除中断。每一个中断都拥有一个屏蔽位。

阻击器 (Blitter) 70 是一个用于快速屏幕刷新和动画的图形处理器，用来作为 CPU30 或 DSP 74 的一个硬件图形例程。它执行 CPU 30 和 DSP 74 写入存储器的指令。它可以通过从系统存储器 33 读取新的命令集来执行任意长度的图形操作序列。它通过阻击器 (blitter) 程序操作变成总线主控设备，因而可以在相当多的周期内独占副系统总线 34。但是，它对于 CPU 30 的优先权并非绝对；在发生一个中断时，它可以被要求放弃副系统总线 34 给 CPU 30。CPU 30 在系统级是优先级最低的总线主控设备；但是它拥有对其它硬件的完全的控制，因而对于副系统总线 34 的使用是在 CPU 30 的程序的完全控制之下的。

阻击器 (Blitter) 有一个多功能比较器以便进行智能的阻击(blitting)操作，还有一个逻辑功能单元(LFU)以便产生输出数据。逻辑功能单元可以多种有用的方式对各数据寄存器的内容进行组合以产生输出数据；比较器可以执行对数据的某种比较以便禁止写操作，并可选择地停止阻击器(blitter)操作。

逻辑功能单元产生输出数据，后者被写入系统存储器 33 中的目的位置。它可以对源和目的寄存器的象素进行任何种逻辑组合。"源数据象素"可以从源数据寄存器或数据模式数据寄存器中选择。LFU 从来自数据寄存器的两组输入数据的四个布尔最小项 (A& B, A& B, A& B, 和 A& B) 中选择一个，并产生所选择的两个布尔最小项的逻辑或。这就允许输入数据的任何逻辑组合；于是存在 16 种可能的操作。

比较器可以对源、目的以及模式数据寄存器中的数据进行多种比较。如果比较条件满足，它便产生一个禁止信号。禁止信号是用来禁止一个写操作的，也可选择地可以停止阻击(blitting)操作。比较器还可以用来提供像素位面 (pixes plane) 效果，给出透明色彩，用于冲突检测和系统存储器 33 的查找操作，以及协助字符绘制。

DSP74 是一个简单的非常高速的用于声音合成的处理器，工作在高达每秒 33 兆指令(MIPS) 的速率。它通过一个 DSP DMA 控制器 (未示出) 访问副系统总线 34，允许它将字或字节读出或写入系统存储器 33。这些传输以短暂脉冲串方式进行，并在 DSP 程序的控制之下。DSP 74 实际上执行的是它自己的私有高速存储器 76 中的程序，并把数据存在那里。

DSP 74 声音协处理器是一个通用的算术协处理器，它有足够的

能力来实现高性能的音乐合成。它提供同步串行输出，用于产生 16 位精度的立体声声音信号，达到通常密致盘技术所具有的声音质量。DSP 74 可由主机 CPU 30 进行微编程，其指令集充分灵活，使得用户可以对设备编程以实现很多与“音乐合成器”大不相同的功能。这些应用可以包括算术语音生成，使用快速傅立叶变换进行音频分析，以及三维图形旋转。DSP 74 使用哈佛体系结构 (Hanard architecture－分离的程序与数据总线) 以达到最大的数据吞吐。DSP 74 有一个算术逻辑单元 (ALU)，其特点是具有 16 位乘以 16 位硬件乘/累加的硬件，以及加，减，和逻辑功能。还有一个分离的串行除法单元，它每个节拍产生一个商数位。

DSP 74 内部的 ALU 是一个 16 位算术逻辑部件，它与本技术众所周知的 Texas Instrument 的74181 具有相同的功能。常见的算术操作被编码为指令；不常见的指令可用通用算术指令 (GAI) 直接设置 ALU 模式位来完成。

DSP 74 有一个 DSP 存储器 76 与之相联。DSP 存储器 76 由程序 RAM，数据 RAM，一个寄存器/常数表，以及一个正弦 ROM 组成 (全都未示出)。总体上说 DSP 存储器 76 既可以在DSP 的内部地址空间也可以在系统存储器 33 的地址空间中访问。DSP 程序 RAM 是 512 个18 位字。这些位置只可以由 CP U 30 写入，对于 DSP 74 来说是程序只读的。程序 RAM 不出现在 DSP 的内部地址空间。当 DSP 74 运行时程序 RAM 不能被主机访问，但在 DSP 空闲时可以访问。

DSP 74 还有一个串行声频数－模转换器 (DAC) 接口。串行 DAC 接口允许 DS P 74 既可以驱动一个同步串行 (I^2S 或者类似的) DAC，也可以从一个同步串行数据源例如CD 驱动器输入数据。

A/V 控制器/协处理器的视频控制器 66 接到外部视频 DAC 50，后者把来自视频控制器66 的 18 位象素信息 78 (红、绿、蓝各 6 位) 转换成一个 RGB 信号 80，这在本技术中是众所周知的。视频 DAC 50 的每一个色彩通道 (R80a, G80b 以及 R80c) 都是通过一个 R2R电阻树和一个 2N2222 晶体管实现的，如图 1C 所示。图 1C 中的设备是如图所示地电路连通的。图 1C 中的电阻 86a－86j 全都是 0.25 瓦的电阻，其值如图所示，容差在 5%之内。晶体管 88 是一个 2N2222。

再次参考图 1B，RGB 信号 80 由一个 NTSC/PAL 编码器 52 转换成一个 NTSC 复合视频信号90。NTSC/PAL 编码器 52 接受色彩时钟 92，HSYNC 和 VSYNC 信号 94—它们是由 A/V 控制器/协处理器 32 的视频控制器 66 产生的，以及红色 80a、绿色 80b、蓝色 80c 视频输出—这些是由视频 DAC50 产生的，然后以著名的 NTSC 格式或基带视频格式产生一个复合视频信号 90。作为替代，也可以产生著名的 PAL (欧洲电视信号标准) 格式。复合视频信号 90 通过一个凹型 RCA 型声音插座 (未示出) 接到外部设备，这在本技术中众所周知。在该较佳实施例中，NTSC/PAL 编码器 52 是一个索尼公司产的 CXA1145。作为替代，也可以使用摩托罗拉公司生产的 MC1377。

声频 ADC/DAC/CODEC 54 使用一条符合著名的菲力普 I^2S 协议的串行链路 96 来链接到 DSP74。ADC/DAC/CODEC 54 把模拟数据转换成数字数据，或者反过来，并且对数字数据压缩和解压缩。A DC/DAC/CODEC 54 把来自可选的麦克风的外部立体声模拟数据 97a－97b 送到 A/V 控制器/协处理器 32。声频输入 97a－97b 使用一个标准立体声 1/4" 连接器接到外部设备。声频

ADC/DAC/CODEC 54 还把来自 A/V 控制器/协处理器的数字数据通过产生左、右声频线输出信号 98a－98b 送到外部设备，例如可选的扬声器 (未示出)，它有两个凹型 RCA 插座，这在本技术中众所周知。如下所述，声频线信号 98a－98b 还被加进 RF 视频信号 22 中。

在该较佳实施例中，ADC/DAC/CODEC 54 是一个由 Crystal Semiconductor 生产的 CS4216。这一部件包含带可编程增益的麦克风输入，以及带可编程衰减的输出。增益和衰减均由 DSP 74 可编程地控制。

作为替代，ADC/DAC/CODEC 54 可由菲力普生产的 TDA1311 DAC 代替。如果使用这种芯片，就得不到 ADC 和 CODEC 功能。

RF 调制器 56 把来自 NTSC/PAL 编码器 52 的复合视频信号 90 与来自声频 ADC/DAC/CODEC 54 的左右声频线输出信号 98a 和 98b 合并到一个载波频率，以产生适合于直接输入到电视机 16 的 RF视频信号 22。为了产生不同的 PAL(欧洲电视信号标准) 和 NTSC 格式，必须使用一个不同的 RF 调制器和晶体。RF 视频信号 22 使用一个凹形 F 同轴连接器接到外部设备，这在本技术中众所周知。

现在参考图 2A－2M 和图 3，示出了本发明的输入设备 18 的一个实施例。如图所示，输入设备 18 由装在同一个壳 100 里的一个触摸板 19 和两只游戏棒 20a 和 20b 构成。壳 100 是用 ABS－T (丙烯腈－丁二烯－苯乙烯三元共聚物，即 ABS grade T，可从 Wong's Electronics Co. Ltd. 公司 (Wong's Industrial Centre, 180 Wai Yip Street, Kwun Tong, Kowloon, Hong Kong) 买到) 做的。输入设备 18接

受模板覆盖 102，每个模板覆盖都是由一个本体 104，一个接头 106，和位于底部的一条边上的一个识别模式 108 组成。模板覆盖102是由一种薄材料做成，例如铜板纸，涂层纸板，或者聚酯胶片。一种合适的聚酯胶片是杜邦公司(Du Pont)生产的很容易得到的“Mylar”牌商标的产品。本体 104 有一幅图形图像刻在里面或者上面；这就是说，图形图案被写，印，漆，雕刻，蚀刻，丝网印刷，或者以其它方法永久性的固定在覆盖体 104 的上面或里面。接头 106 从本体 104 伸出，用来扣住模板覆盖 102。识别模式 108 将在下文结合图 2G－2K 来描述。

触摸板 19 有一块板面 110，露在外面以便用手指、铁笔 21 或者类似的东西接触。板面里面或上面刻有一幅缺省模板图形图案；就是说，缺省模板图形图象被永久性地写、印、漆、雕刻、蚀刻、丝网印刷或以其它的方法固定在板面 110 上或里面。

如图 2A 所示，刻在板面 110 里或上面的缺省模板图形图案所支持的功能如下：“enter (进入)”，“exit (退出)”，“pause (暂停)”，“previous (上一个)”，“next (下一个)”，以及箭头键(上、下、左、右)。作为替代，可用“select (选择)”代替“enter (进)”，用“cancel (取消)”代替“exit”。作为另一种替代，可以绘出十个矩形区域－每一个表示为一个阿拉伯数字。作为又一种替代，也可将每个英文字母映射到触摸板的一个区域。作为又一种替代，可以把一个 QWERTY 键盘映射到板面 110 上。确实，实际上可以选择任何模式，或者模式与符号的组合。应该选择缺省模板图形图象使得它适用于大量的用在系统 10 上的应用程序。

输入设备 18 包含两种形式的夹持器以固定模板覆盖 102 贴紧板面 110：(1)一个凸唇112，它由三个直唇部分 112a－c 组成，形成一

个大体呈 U 型的槽，从三面固定模板覆盖102；(2) 一对隆起 116a和116b，它们在第四边固定模板覆盖 102。凸唇 112、槽 114，和隆起 116a，116b 在图 2C，2D，2E 中详细示出，并在伴随这些图的文字中加以描述。

一个用于拿住输入设备 18 的手柄 118 和一个用来放置铁笔 21 的圆孔 120 也在图 2A 中示出。

现在参考图 2B，显示了输入设备 18 有一个模板覆盖 102 紧贴在板面 110 上的一个顶视图。如图所示，覆盖 102 的本体 104 的三个边插在三个凸唇 112a－112c 的下面。固定覆盖 102 的本体 104 的第四条边的隆起 116a 和 116b 也显示在图中，在覆盖 104 的接头106的两边各有一个。覆盖 102 是这样被插入的：把覆盖 102 的本体 104左手边及右手边插到凸唇部分 112a 和 112c 的下边，把覆盖 102 向下滑动，直到覆盖 102 的本体 104 的底边位于另一凸唇 112b 部分的下面。最后放开模板 102，接头 106 被夹在隆起 116a 和116b 之间，后者固定了覆盖的顶边。

图 2C，2D 和 2E 显示了隆起 116a、116b 和覆盖 102 的细节。图 2D 显示了接头 106 延伸到隆起 116b 的外边。图 2E 显示了覆盖 102 的本体 104 紧靠隆起 116b。

覆盖 102 是这样移去的：用拇指和食指捏住接头 106，抬起覆盖 102 的本体 104 到隆起116a，116b 以上，这样使覆盖 102 可以从 U形唇 112 之下越过隆起 116a 和 116b 而滑出。

图 2F 显示了从三面固定覆盖 102 的槽 114。该图中还示出了触摸板传感器 122，一个覆盖传感器 124，一个用来放置多个模板覆盖 102 的空腔 126，以及一个充分刚性的基座 127，它用与壳 100 相同

的材料制成，提供一个充分的阻力，使得对传感器 122 的接触能被检测到。

触摸板传感器 122 的位置紧贴所述板面 110，它是这样配置的：使得手指、铁笔或类似物体对板面 110 上面或附近施加的压力能够让传感器 122 检测到所触的位置。

触摸板传感器 102 可以是许多类型中的一种，例如基于阻抗的传感器，声学传感器，和开关闭合型传感器。其实例包括薄膜开关矩阵，(例如授予 Florella 的美国专利第 4, 736, 190 号所公开的设备)，和高分辨率开关，闭合型传感器 (例如授予 Ito 等人的美国专利第 4, 529, 959 号所公开的设备)。一种合适的传感器 102 可以从 Wong's Electronics Co. LTD (Wang's Industrial Centre, 180 Wai Yip Street, Kwun Tong, Kowloon, Hong Kong) 买到，即部件号 PR39983。

传感器 122 至少产生一个电信号来响应铁笔 21、手指或类似物体对板面 110 的接触或者对紧贴板面 110 的模板覆盖的接触。例如：开关闭合型传感器一般需要一定数量的驱动器来顺序地驱动一个轴上的许多条线以及一定数量的接收器以检测另一个轴上的许多条线中的哪一条正在传导由驱动器驱动的信号。知道了哪个接收器检测出了哪个驱动器产生的信号使得可以确定造成开关闭合的触摸的位置。

覆盖传感器 124 是一个相应覆盖 102 上的识别模式 108 而产生一个电信号的传感器。这样触摸板 19 的覆盖传感器 124 和覆盖模板 102 的识别模式 108 必须在位置上和设备技术上都相一致。一个合适的覆盖传感器 124 是一排六个光电发射器/接收器，每一个都有一个有一定角度的发射器和一个有一定角度的接收器，它们在本技术

中众所周知，并可以从 Wang's Electronics Co. LTD 买到，即部件号 PR39990。

如图 2G 所示，覆盖传感器 124 通过壳 100 上的六个孔 128a－128f 与识别模式 108 光学耦合。在每一个孔 128 下面是一个光电发射器/接收器对（未示出）。在另一个实施例中（未示出），覆盖传感器 124 的六个发射器/接收器对和六个孔 128a－128f 可以被分成两组，每组三个，每组分别放在触摸板传感器 122 的两边。这就是说，三个孔 128a－128c（以及它们的相关的发射器/接收器对）可以位于一个凸唇部分 112a 的下边，而另外三个孔 128d－128f（以及它们的相关的发射器/接收器对）可以位于另一个凸唇部分 112c 的下面。

图 2H－2K 显示了识别模式 108 的几个例子。识别模式 108 位于模板覆盖 102 的一条边上。有六个独立的记号—每个对应于一个覆盖传感器 124 的一个独立的发射器/接收器对。当覆盖 102 在位时，识别模式 108 与传感器 124 对齐。如果使用上面描述的、传感器有两组各三个孔的替代实施例，那么识别模式 108 必须同样地分成两组各三个，并把两组分别置于触摸板传感器 122 的两边。

使用光电发射器/接收器对作为传感器 124 使得可以在模板覆盖 102 里设计非常简单的识别模式 108。如果制造覆盖的材料是白色的，那么在覆盖背面的涂有黑墨或颜料的区域可以用来作为识别模式的一种形式，而没有黑墨的区域可用作另一种形式。

图 2H－2K 中显示了几种不同的识别模式的可能组合。图 2H到 2K 显示的识别模式 108 分别对应于二进制模式 010001_2，011110_2，000000_2，111111_2。这样，识别模式看起来是沿着模板覆盖 102 的边缘

分布的一组明暗区域。模式 000000_2 只是为了说明的目的而示出的。在实际应用中，所有的 000000_2 的全白的模式都不可能被使用，因为这种模式对应于任何模板覆盖都不存在，在这种情况下会使用缺省的模板图案。

图 2L 是本发明的输入设备 18 的一个前部正视图，它示出了把游戏棒 20a、20b 分别置于触摸板 19 的两侧。如本图所示，游戏棒 20a 和 20b 各有一个瞬时按钮开关 130a、103b 分别固定在它们各自的端部。图 2L 还显示了开向空腔 126 的矩形孔 132，空腔 126 是用来容纳多个模板覆盖的。

图 2M 是沿由图 2A 中的线 2M－2M 所标明的平面所截的部分剖面图，它示出了用于本发明的输入设备的游戏棒的细节。游戏棒 20a 和 20b 的细节在图 2M 中被重复示出。另外，图 2M 的细节是 90° 旋转对称的；这样，虽然许多结构中的两个显示在该图中，实际上在这一特定实施例中使用了四个同样的结构。游戏棒 20a 固定在一个棒 134 上，后者穿过一个孔 136 伸进壳 100 所定义的空间。棒 134 止于一个支点 138，后者在开关底座 140 上转动。开关底座 140 是通过在几个固定器 14 4a－144d 的四个螺丝 142a－142d 固定在壳 100上的，支座是物理地附在壳 100 上的。孔 136 是由一个环形封圈 146封住的，封圈的环整齐地围住棒 134，封圈的外缘通过普通的方法在孔内物理地附在壳 100 上。

游戏棒 20a 关于支点做 14 度的自由运动；就是说，棒 134 可以实际上向各个方向偏移垂直于由开关底座 140 定义的平面的方向七度。棒 134 有四条传动臂 150a－150d 物理地附在上面。传动臂 148的位置紧贴四个橡皮圆顶瞬时开关按钮 150a－150d，后者物理地附在

开关底座 140 上。臂 148 和开关 150 是这样配置，使得当游戏棒偏离垂直位置时，一个或多个臂 148 产生与它们相连的开关的一次闭合事件。这样，游戏棒 20a 的运动就被开关 150 的闭合事件检测到了。作为替代，游戏棒 20a，2 0b 可以用其它结构实现，例如基于电位计的系统，这在本技术中众所周知。

现在参考图 3，它示出了输入设备 18 内部电路的方框图。输入设备 18 包含触摸板传感器 122，覆盖传感器 124，游戏棒传感器 200a，200b，坐标确定电路 202，覆盖检测电路 204，一个 100 毫秒定时器 208，以及接口电路 210，所有设备都如图 3 所示电路连通。

坐标传感器 122 和覆盖传感器 124 如上面结合图 2 的文字所描述。坐标确定电路 2 02 与坐标传感器 122，接口电路 210，和 100 毫秒定时器 208 电路连通。坐标确定电路 202 设置为接受来自坐标传感器 122 的电信号 203 并确定对应于手指、铁笔 21 或类似物体所触位置的 X 轴和 Y 轴的值。例如，如果坐标传感器 122 是一个开关型传感器，那么坐标确定电路 202 将包含驱动器和接收器来确定哪个开关闭合了，这在本技术中众所周知，并包含把那个开关的位置翻译成相对于板面 1 10 的一个有意义的值的逻辑。

覆盖检测电路 204 与覆盖传感器 124 和接口电路 210 电路连通。覆盖检测电路 204 接收来自覆盖传感器 124 的电信号 205 并产生一个相应于识别模式 108 的消息，或者相应于没有识别模式的消息，即如上所述检测到 000000_2。

方向确定电路 206 与游戏棒传感器 200a，200b 和接口电路 210 电路连通。游戏棒传感器 200a，200b 包含四个橡皮圆顶开关 150a－150d 和两个游戏棒开关 130a，130b，如前所述。方向确定电路

产生一个基于这些开关的闭合事件的消息。

100 毫秒定时器 208 与坐标确定电路电路连通。定时器 208 反复确定 100 毫秒时间周期的到时并产生一个信号 209 来指示该时间周期的到时。坐标确定电路 202 使用该信号 209 来检测在由定时器 116 检测到的 100 毫秒到时信号之间，手指、铁笔 21 或类似物体所触位置的改变。

接口电路 210 与坐标确定电路 202，覆盖检测电路 204，数据处理单元 12 (通过串行数据链路 22)，以及 (如果有的话) 其它输入设备，通过串行数据链路扩展 23 电路连接。接口电路 210 接受由坐标确定电路 202 确定的坐标值，由覆盖检测电路 204 产生的覆盖消息，以及由方向确定电路 206 产生的消息，并把任何这样的消息通过串行数据链路 22 发送给数据处理单元 12。

所有的输入设备都与处理单元 12 成菊花形链接。这样，接口电路必须传递来自其它输入设备的信息包到 CPU 30。下面将会详细解释，每一个接到处理单元 12 的输入设备都有一个唯一的设备号。离处理单元 12 最近的设备的设备号是 0，一个设备离处理单元 12 越远，它的设备号就越高。但是，输入设备并不知道它们自己的或其它设备的设备号。这样，每个设备必须给来自同类型其它设备的数据包的设备号加 1。在链中任何设备号大于 15 的设备都被忽略。

例如，假设有三个同样类型的输入设备 α、β、γ，它们是这样接到处理单元 1 2 的：α 接到处理单元 12，β 接到 α，γ 接到 β。因此，α 的设备号是 0，β 的设备号是 1，γ 的设备号是 2。其它设备并不知道它们自己的或者其它设备的设备号。每个设备都用设备号 0 发送自己的数据包。

当 α 把一个数据包传送给处理单元 12 时，缺省设备号 0 是正确的，因为 α 是最接近处理单元 12 的。但是，β 和 γ 同样用设备号 0 发送数据包。为了解决这个问题，每个设备给通过它传递的数据包的设备号加 1。这样，当 β 把一个来自 γ 的数据包传给 α 时，β 给设备号加 1，这样来自 γ 的包设备号就是 1。同样，当 α 把来自 γ 的数据包传送给处理单元 12 时，α 给其设备号加 1，因而给了来自 γ 的包一个正确的设备号 2。这样，链中的每一个设备给来自同类型的其它设备的每一个数据包的设备号加 1，并把它传给下一个设备。

因此，除了传递从其它输入设备收到的数据包之外（如果有的话），接口电路 210 还给通过串行数据线扩展 23 收到的来自其它同类型设备的任何数据包的设备号加 1。接口电路 210 把带有修改过和没修改过的设备号的数据包传递给数据处理单元 12。

具有本发明的输入设备 18 的系统 10 非常易于使用。输入设备通过串行链路 22 发送数据包给数据处理单元 12。如前面所提到的，输入设备通过 I/O 协处理器 36 与 CPU 30 相连。每个输入设备与下一个输入设备成菊花形链接。I/O 协处理器 36 以先入先出 (FIFO) 方式接受数据包并存储它们。

每 50 毫秒“滴答” I/O 协处理器中断一次 CPU 30。作为响应，CPU 访问协处理器 36 的I/O 端口 AS0 的一个字节以决定自上次 CPU 访问以来“滴答”的次数以及需要传输的设备消息的数目，如前所述。设备信息的十种类型如下表所示。

设备类型	字节 0 (计数)	字节 1(设备 ID)		字节 2	字节 3	字节 4
		(位 4 - 7)	(位 0 - 3)			
键盘 (PS/2)	2	0	0	扫描码	不用	不用
鼠标器 (PS/2)	4	1	0	鼠标字节 1 (按钮)	鼠标字节 2 (X 数据)	鼠标字节 3 (Y 数据)
开关闭合	1-255 变量	链中设备号	1	按钮状态 0: 开; 1: 关	按钮状态 (可选)	按钮状态 (可选)
游戏棒 (数字)	2	链中设备号	2	开关关/开代码	不用	不用
坐标 (相对)	4	链中设备号	3	按钮状态	X 增量	Y 增量
坐标 (绝对)	4	链中设备号	4	按钮状态	X 坐标	Y 坐标
触摸板覆盖消息	2	链中设备号	5	覆盖代码 (0-63)	不用	不用
动作消息	1-255 变量	链中设备号	6	动作按钮数据	(可选)	(可选)
系统穿过	1-255 变量	链中设备号	14	字节 0	字节 1 (可选)	字节 2 (可选)
初始化消息	Variable 1-255 变量	链中设备号	15	设备类型 (ID)	厂商代码字节 0	厂商代码字节 1 (可选)

从表中可以看出，消息结构长度不同并且其结构与所对应的设备类型密切相关。对于从各个 I/O 设备发往 I/O 协处理器的数据和由 I/O 协处理器发往 CPU 的数据来说，表中的设备消息是相同的。除了上表所示的结构之外，从 I/O 设备发往 I/O 协处理器的每一条消息都有一个校验和，以保证从输入设备 18 发往处理器单元 12 的数据正确无误。校验和是一个标准的模 256 校验和，其中校验值是把

所有字节相加的和为 0 所需的值 (相加时忽略进位)。I/O 协处理器先去掉该校验和然后把数据发往 CPU。因此, CPU 读出的字节流实际上与 I/O 协处理器收到的字节流是相同的, 只有下列例外: (1) CPU 读出的第一个字节是一个特殊字节, 包含滴答数及 I/O 消息的数目; (2) 丢失了校验和。

PS/2 鼠标器和键盘作为 0 型设备被支持。键盘的链号为 0, 鼠标器的链号为 1。这些设备由 I/O 协处理器使用现有的 PS/2 协议通过串行数据链路来支持。

设备类型 1 是用于有多个按钮的设备的。最多到 255 个字节 (每字节 8 个按钮) 或者说 2040 个按钮可以用这种消息类型输入到系统中。开的按钮作为逻辑 0 发送而合的按钮作为逻辑 1 来发送。这是一个可变长度消息。

数字游戏棒, 例如游戏棒 20a 和 20b, 作为 2 型设备被支持。每一个触摸板 19 有两个游戏棒与之相联。每一个游戏棒有一个唯一的链号。每个左游戏棒的链号是奇数 (1, 3, 5, 7, 9 等), 而每个右游戏棒是偶数 (0, 2, 4, 6, 8 等)。每一个游戏棒独立地报告。该消息是一个定长消息。回忆一下数字游戏棒传感器包含一定数目的开关 150a－150d。该消息是一个最多代表 8 个开关的字节, 它包含了运动传感器开关150a－150d 和数据输入开关, 例如开关 130a。这一类型的消息字节的各个位分别代表: 向上开关 (最高位), 向下开关, 向左开关, 向右开关, 开关# 1, 开关# 2, 开关# 3, 和开关# 4 (最低位)。触摸板 19 所带的游戏棒 20a 和 20b 只有一个按钮 130, 它对应于上述开关# 1。其它三个按钮永远报告为 0。

坐标型设备例如鼠标器和跟踪球作为 3 型设备报告。紧跟 ID 的

第一字节报告设备的任何按钮信息。最多可报告 8 个按钮。下一字节是 X 增量值，再后是 Y 增量值。X 和 Y 的增量值是基于上次报告的设备位置的。如果需要，应用程序必须把这些值转换为绝对坐标。最大运动为 255。如果实际运动超过 255，则将发送两个或更多的消息。这是一个定长消息。

触摸板 19 作为 4 型设备被支持。这一类型的其它设备包括模拟游戏棒。紧跟 ID 的第一字节是用来报告按钮信息。而下一字节是用于报告绝对 X 位置的。依次而后的是绝对 Y 位置。绝对 X 值和 Y 值都是 1 个字节并且限制在 0－255 范围内。这是一个定长消息。

触摸板覆盖 102 是作为 5 型设备报告的。触摸板覆盖是使用触摸板上的 6 位传感器 124 检测的。当触摸板测到一个覆盖的改变时，便产生一个消息。所有的覆盖代码都依赖于应用程序，应用程序必须认识每个覆盖的代码。这是一个定长消息。

动作消息是用来定义预定义的独立于设备的功能的公用集合的。这些功能可以由多种类型设备以不同方式产生，但是由系统和应用程序以同样方式使用和翻译。动作消息作为 6 型设备使用一个变长消息来报告。在此特定实施例中，定义了三种独立于设备的功能，并分别与该字节的低三位相关联：START (开始—开始一个活动或进程) PAUSE (暂停—暂停一个活动或进程)，SELECT (选择—选择多种事件或动作中的一个)。设置相应位以报告这些功能。所有的其它位都被保留以备将来之用，并且向 CPU 报告为 0。

系统穿过消息类型是用来处理不适用于任何一个前面定义的设备类型的设备类型的。它使用消息类型 14。这是一个变长消息。对其数据的定义是依赖于设备的，并且是应用程序专用的。每个应用程序

必须把这一类型的消息翻译成所需的功能。

来自每一个设备的第一条消息是 15 型设备。它用来告诉系统一个设备将要发送输入消息了。该消息还定义了将被用来报告输入的未来的设备类型。这是一个变长消息。

" 在系统加电和 50 毫秒间隙时 I/O 协处理器察看盒式存储器和扩展传感线以确定配置，并提醒系统，还发送一个配置字节给 CPU。这是在系统加电时 CPU 从 I/O 协处理器收到的第一个字节。当检测到发生一个变化时，I/O 协处理器只产生一个模块配置中断，盒式存储器状态的一个变化造成一次系统复位，因而造成 I/O 协处理器向 CPU 发出另一个配置字节。设置所发送字节的适当的位来指示相关项的存在：位 0 对应于盒式存储器 1，位 1 对对应于盒式存储器 2，位 2 对应于可选的 CD 驱动器。其它位被置为 0。

另外，CPU 能通过把信息沿着串行链路 22 写入 I/O 协处理器 36 来发送数据给 I/O 设备。写入 I/O 端口 AS0 的数据字节中每个字节前面都有字节 03H。I/O 协处理器把这些字节写入 I/O 设备。这种功能是用于发送数据给例如一个打印机的(未示出)。

与这种具有两个游戏棒和一个带有刻上的缺省模板图形映象的触摸板的本发明的输入设备的连接也是很简单的。来自系统 BIOS 并在 CPU 30 上执行的一个中断服务程序通过 I/O 协处理器 36 从输入设备接收数据。中断服务器仅把来自 I/O 协处理器 36 的传送数据放进存储器 33 中。运行在 CPU 上的应用程序周期性地通过一个软件中断来查询操作系统 BIOS 以确定是否收到了输入。如果是，它们就响应于软件中断通过操作系统与应用程序通信。

应用程序监视当前的模板。如果检测到缺省的模板图形映象(覆

盖传感器 124 会检测所有的发送型识别模式 108, 即 000000_2), 那么应用程序相应于缺省模板操作。另一方面, 如果检测到一个模板覆盖 102, 那么应用程序相应于该特定模板覆盖操作。

CPU 30 上运行一个对与触摸板的接口提供特殊支持的操作系统。操作系统或者从系统存储器 33 和 ROM 中装入, 或者从盒式存储器 ROM 40 中装入。该操作系统具有下列命令可以被执行在 CPU 30 上的应用程序所调用: "define_a_region (定义一个区域)", "set_mapping_units (设置映象单位)", "clear_all_region (清除所有区域)", 以及"interpret_a_point (翻译一个点)"。

"define_a_region" 命令允许应用程序定义触摸板的一个区域并把该区域与某一个区域标识符("region_id") 相联系。该被接触的定义区域内的任何一点都用这个 region_id 标识。在一个实施例中, 在初始化后, 整个触摸板被定义为一个无效区域 (null region), 它具有一个无效区域标识符 ("null_id")。就是说, 任何 null_id 位置被触都不会导致操作系统执行任何任务或功能, 即没有参数传给应用程序。作为替代, 触摸一个无效区域可以触发一个错误处理例程, 后者可能比如让系统用发声或"蜂鸣"的方法指示用户触了一个无效区域。"define_a_region" 命令把所选形状的区域赋予其它 region_id 的区域。

实际上任何形状或形状的组合都可以用 "define_a_region" 命令来定义: 圆, 矩形, 三角形等。另外, 多个形状可以合并形成一个复杂形状区域。例如, 五个三角形可以和一个正五边形组合形成一个星形。另外, 无效区域可以用 "define_a_region" 命令包括进来, 这样允许定义空心形状, 例如环。

"set_mapping_units" 命令允许应用程序为 "define_a_region" 命

令定义不同单位，因而允许操作系统支持不同分辨率的触摸板。例如，操作系统缺省为矩形设备分辨率单位，这是由行和列的分辨率(宽度和间距)确定的；通过使用"set_mapping_units"命令，应用程序可以使操作系统把单位改为比如毫米或百分之一英寸。另外，应用程序可以把单位设置为其它的值。例如，如果系统显示设备的分辨率为640象素×480象素，应用程序可以用"set_mapping_units"命令给触摸板单位赋予640×480相应于象素的单位。

"clear_all_regions"命令取消所有的以前用"define_a_region"命令定义的区域，实际上把整个触摸板定义为一个无效区域。当有一个新的模板覆盖放到触摸板面上时，应用程序调用该命令，于是清除了与前一个模板覆盖关联的所有定义区域(如果有的话)。在用"clear_all_region"命令清除了区域后，应用程序调用"define_a_region"命令定义新模板覆盖的区域。

"interpret_a_point"命令使得操作系统确定所触区域的region_id。如果是操作系统在监视触摸板的触摸，该命令由操作系统自己启动；如果是应用程序在监视触摸板的触摸，则该命令由应用程序启动。在一实施例中，应用程序查询操作系统看操作员是否触摸了一个触摸板区域。

每次触摸板被触，触摸板便发送设备专用坐标信息给CPU。"interpret_a_point"命令的一部分是由CPU确定所触区域。确定以后，操作系统向应用程序返回一个代码，后者或者是对应于所触区域的"region_id"，或者是指示没有区域被触的代码。

在操作系统中支持触摸板的计算机系统的使用十分简单。首先，一个应用程序定义触摸板区域并用"define_a_region"命令把区域与

region_id 相关联。每一个被定义的区域都必须用 "define_a_region" 命令定义;任何未定义的区域仍为具有 null_id 的无效区域。

随后对触摸板面的触摸或者由操作系统或者由应用程序检测,这取决于谁在监视触摸板。如果操作系统在监视触摸板,它检测触摸并自己启动 "interpret_a_point" 命令以确定所触区域的 region_id。如果 region_id 是 null_id,操作系统执行一个适当的动作,例如蜂鸣。在此情况下,它不会向应用程序传递任何参数。如果 region_id 不是null_id,操作系统把与所触区域关联的 region_id 传递给应用程序,由应用程序执行与所标识的 region_id 相关联的功能。如果应用程序在监视触摸板并检测触摸板的触摸,其过程是相同的,只有一点不同即应用程序导致操作系统启动 "interpret_a_point" 命令而不是操作系统自己启动之。

如果一个新的模板覆盖放在触摸板表面,应用程序便调用 "clear_all_regions" 命令,后者使操作系统再次清除所有定义区域并认定整个触摸板面为一个无效区域。应用程序再次调用 "define_a_region" 命令以使操作系统定义该模板的适当区域并在应用程序指导下给它们赋予 region_id。

一个定义或映射所定义形状的区域的特定例子在图 4 描绘出来。它显示了一个被映射的触摸板。敏感区域的形状为 (1) 一个箭头,用 196 标出; (2) 一个圆和一个环,由一个环形无效区域分开,总体用 198 标出。敏感区域各自由小的离散符号示出,下面将加以识别。在初始化后或执行 "clear_all_regions" 命令后,触摸板的所有敏感区域被赋予 null_id,后者在图 4 中用符号“·”表示。这意味着如果一个“·”离所触位置最近,那个区域便被赋予 null_id,操作系统如上文

所述地响应。

为了创建被一个环形无效区域分开的圆和环，应用程序三次调用 "define_a_region" 命令。首先，应用程序调用命令 "define_a_region" 时参数为 (circle, 14, F, 4, 1)。“circle”指示要画的形状；“14, F”指示圆心坐标，“4”指示以当前单位表示的圆半径，这里的单位是矩形设备分辨率单位；“1”指示赋予此圆的 region_id。这一命令创建了一个虚拟圆 200，在图 4 中它围住了由圆 200 包围的 52 个敏感区。这 52 个敏感区的 region_id 都被赋予 1，并在图 4 中用下列符号“⊙”、“·”和“○”指出。28 个用“⊙”符号指出的敏感区的 region_id 仍被赋予 1；其余的将被后面的“define_a_region”命令“重赋值”为其它的 region_id。

其次，应用程序调用“define_a_region”命令时参数为 (circle, 14, F, 3, 0)，其中“0”指示 null_id 作为 region_id。它创建了一个虚拟圆 202，它围住了由圆 202 包围的 24 个敏感区。这 24 个敏感区的 region _id 全都赋予 0，并在图 4 中用下列符号指示：“·”和“○”。回忆一下这 24 个敏感区曾被前一个“define_a_region" 命令赋予 region _id 为1。这样，这些 region_id 也可以认为是被“重赋值”为 0。12 个用“·”示出的敏感区的 region_id 将仍被赋予 0；12 个用“○”符号指出的敏感区将由第三个 “define_a_region” 命令 “重赋值” 为另一个 region_id。12 个用“·”符号指出的敏感区是无效区，如上所述。

最后，为完成这一用一环形无效区域分开的圆和环，最后一次调用命令“define_a_region”时参数为 (circle, 14, F, 3, 2)。该命令创建一个虚拟圆 20 4，它围住了用“○”符号指示的 12 个点。这 12 个敏感区的 region_id 被赋予 2。回忆一下这 12 个敏感区曾被第一个“define_

a_region"命令赋予 region_id 为 1，又曾被前一个"define_a_region"命令"重赋值"为 null_id。这样，这些区的 region_id 也可以认为是被"重赋值"为 2。

最终得到的复杂形状 198 是由一个环形无效区域 210 (region_id 赋为 0)分开的一个 region_id 赋予 2 的填充 206 和 region_id 赋为 1 的环 208。填充圆 206 中的敏感区被用符号"○"表示。环 208 中的敏感区用符号"⊙"表示。环形无效区域 210 中的敏感区用符号"·"表示。

作为替代，"define_a_region"命令可以配置为直接支持一个环，这样复杂形状 198 可以用两个命令创建：一个圆和一个环。

创建箭头 196 需要两次调用"define_a_region"命令。第一次调用参数为 (rectangle, 6, C, 9, I, 3)，其中"vectangle"指示形状，"6, C"指示矩形左下角，"9, I"指示矩形右上角，"3"指示 region_id。这一命令创建一个虚拟矩形，它包围用符号"■"指示的敏感区。第二个"define_a_region"命令的参数为 (triangle, 1, F, 6, A, 6, K, 3)，其中"triangle"指示形状，"1, F"，"6, A"，"6, K"指示三个顶点，"3"指示 region_id。这一命令创建一个虚拟的三角形，它围住用符号"▲"指示的敏感区。注意矩形 212 和三角形的 region_id 都赋予 3；它们有相同的 region_id。这样，这两个形状映射到同一功能并组成箭头 196。正因如此，接触任何一个区域或两个区域都接触都使应用程序产生同样的反应。触摸板表面的其它敏感区，总体用 215 表示，其 region_id 仍被赋予 null_id。这些敏感区用符号"·"指出。

显而易见，大量的复杂形状可以用这种方法形成。每一个形状有一个唯一的 region_id 或者一个或多个区域共享一个 region_id。

在使用中，按压图 4 所示触摸板的映射区域会导致操作系统判断所触区的 region_id。例如，如果按压了位置 13, E(由 216 指出)，所触位置的坐标通过串行链路 22 传给 CPU，操作系统判断 region_id 2 被压，并把 region_id 2 传给应用程序，后者于是执行与 region_id 2 关联的功能。

另一方面，按压赋予 null_id 的区域不会造成操作系统把一个 region_id 传给应用程序。而是如上所述，操作系统或者什么也不做或者蜂鸣或者执行其它适当的动作。为了让应用程序检测接触一个“无效”区域，应用程序必须先调用“define_a_region”命令给整个触摸板表面赋予某一个 region_id，并在制做空心区域时，给任何空心区域赋予那个 region_id。

在上面的映射中术语“虚拟”被用来描述这些形状：圆 200，圆 202，圆 204，矩形 212，和三角形 214。使用这个术语是因为敏感区域并不具有无限的分辨率。实际上，当使用由不同符号定义的术语“敏感区”时，该术语意思是表面上离那些符号最近的区；因此，任何定义的形状都将是映射出的敏感区的区域形状的近似。触摸板的分辨率越高，映射的敏感区就越接近于定义它们的虚拟形状。

虽然本发明是通过对它的实施例的描述加以说明的，虽然这些实施例被相当细致地描述，申请人的目的并不是要限制或以任何方式局限所附权利要求到如此的细节。其它的优点和修改对于这方面的技术人员来讲是显而易见的。因此，本发明在广义上并不局限于特定的细节，代表性设备和方法，以及所示和所述的说明例。因此，可以脱离这些细节而并不背离申请人的总体的创造性原理的精神或范围。

说 明 书 附 图

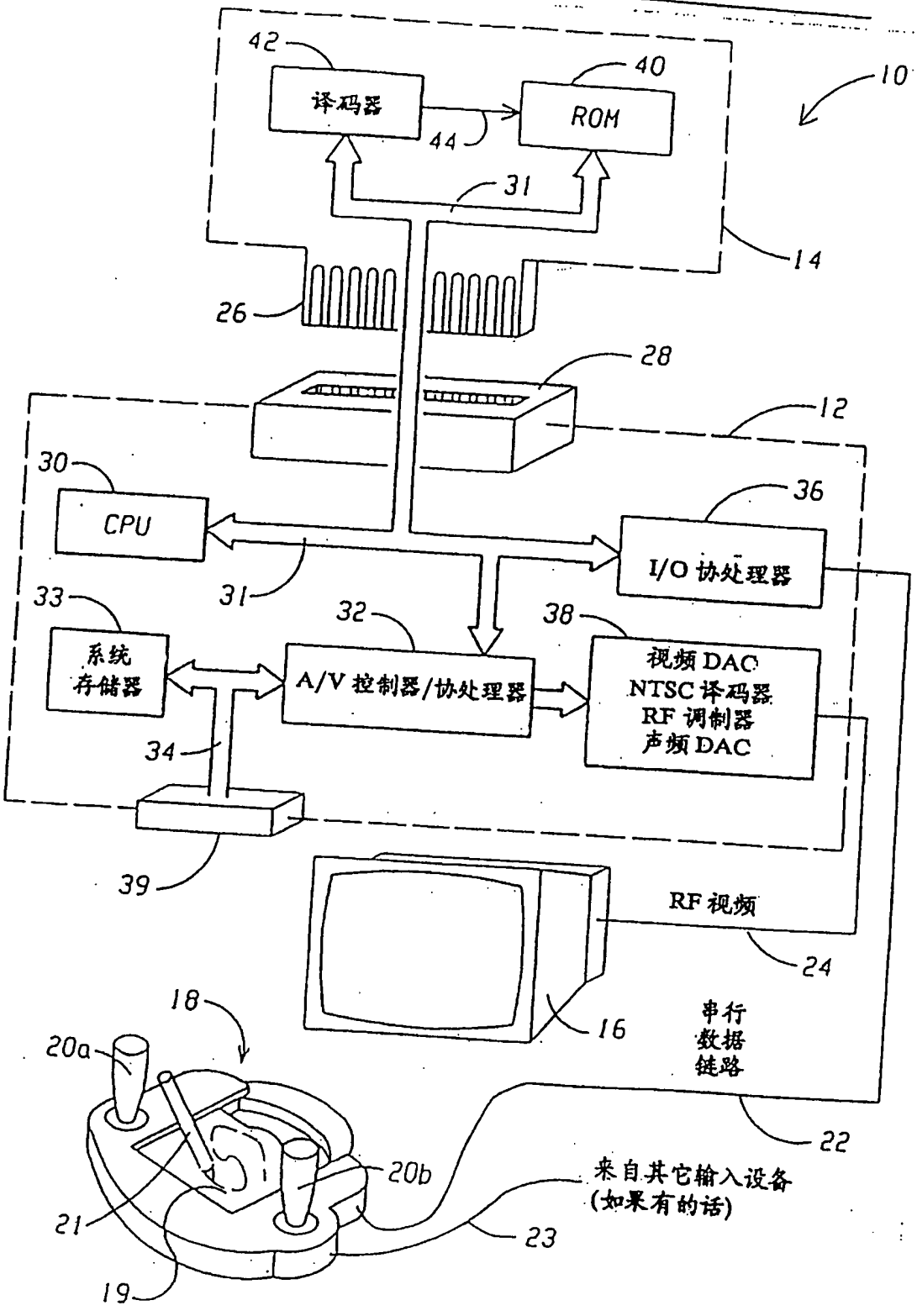

42
译码器
40
ROM
10
44
31
14
26
28
12
30
CPU
36
I/O 协处理器
31
33
32
38
系统
存储器
A/V 控制器/协处理器
视频 DAC
NTSC 译码器
RF 调制器
声频 DAC
34
39
RF 视频
24
18
16
串行
数据
链路
20a
22
20b
来自其它输入设备
(如果有的话)
21
23
19

图 1A

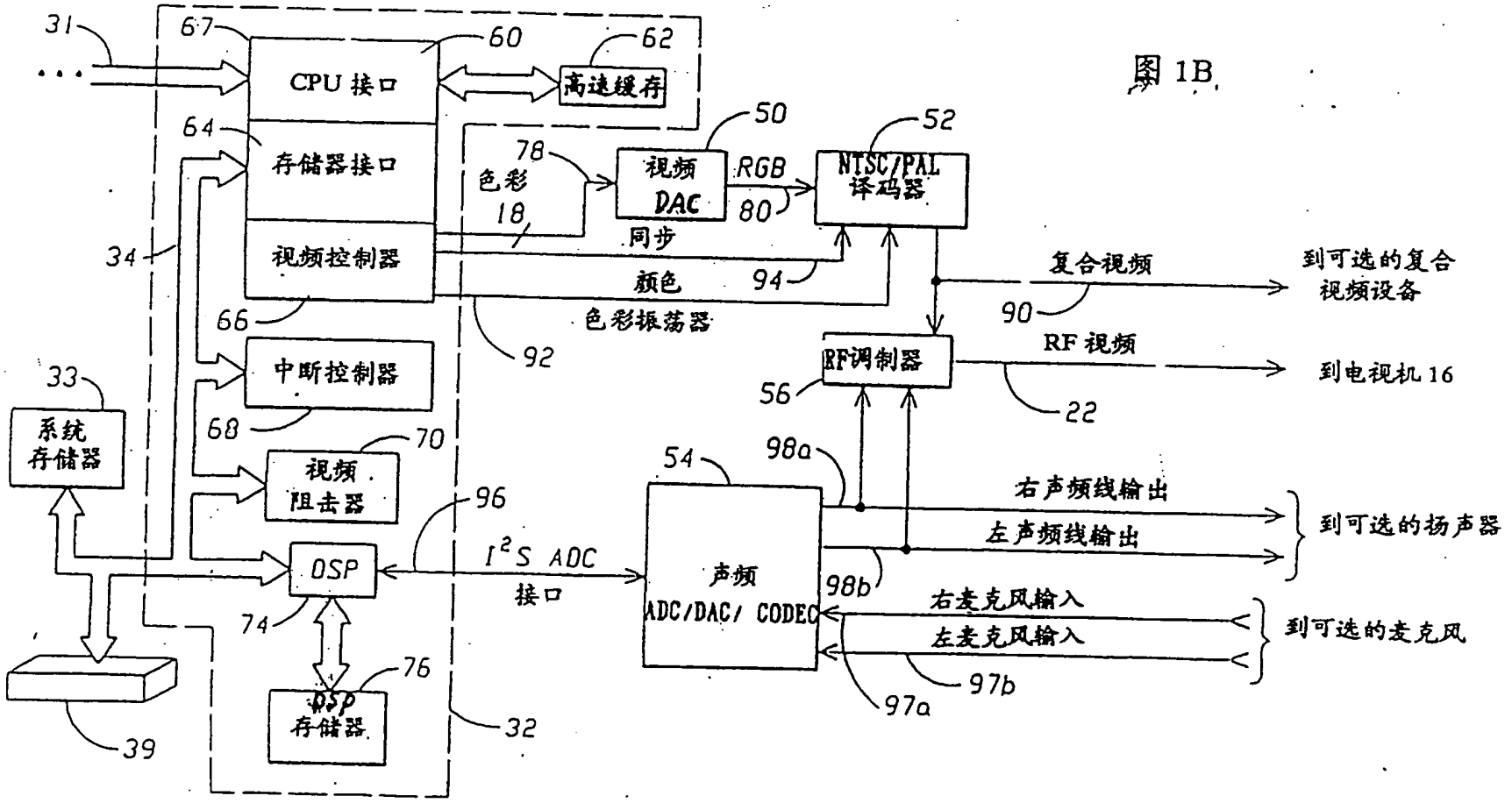

31
67
60
62
CPU 接口
高速缓存
64
存储器接口
50
52
78
色彩
18
视频
DAC
RGB
80
NTSC/PAL
译码器
同步
34
视频控制器
颜色
94
复合视频
到可选的复合
视频设备
90
色彩振荡器
66
92
RF 视频
RF调制器
到电视机 16
56
22
33
中断控制器
68
系统
存储器
70
视频
阻击器
54
98a
右声频线输出
96
左声频线输出
到可选的扬声器
DSP
I²S ADC
接口
声频
ADC/DAC/ CODEC
98b
右麦克风输入
左麦克风输入
到可选的麦克风
74
97b
97a
76
DSP
存储器
32
39

图 1B

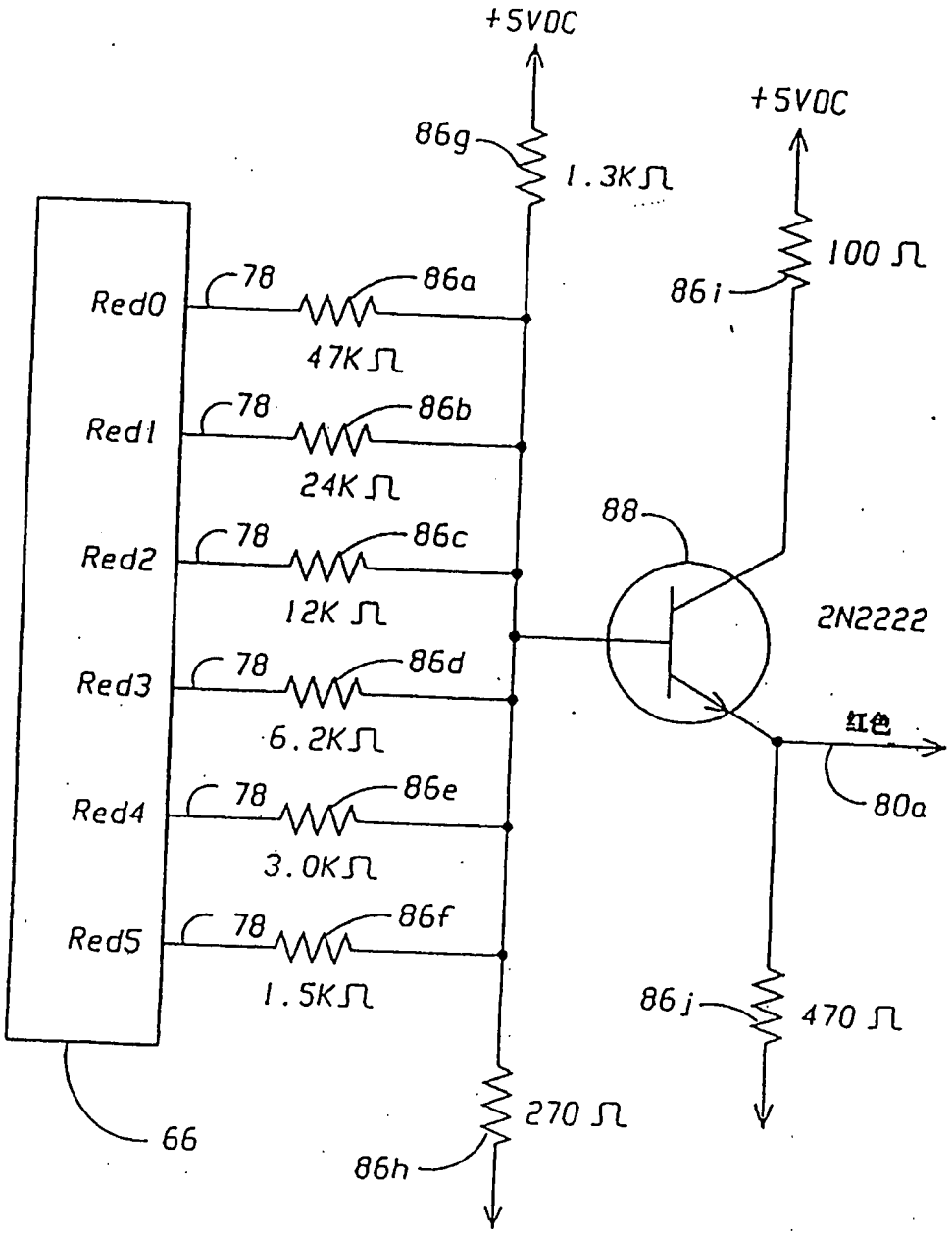

+5VDC
86g
1.3KΩ
+5VDC
100Ω
86i
Red0
78
86a
47KΩ
Red1
78
86b
24KΩ
88
Red2
78
86c
12KΩ
2N2222
Red3
78
86d
6.2KΩ
红色
80a
Red4
78
86e
3.0KΩ
Red5
78
86f
1.5KΩ
86j
470Ω
66
270Ω
86h

图 1C

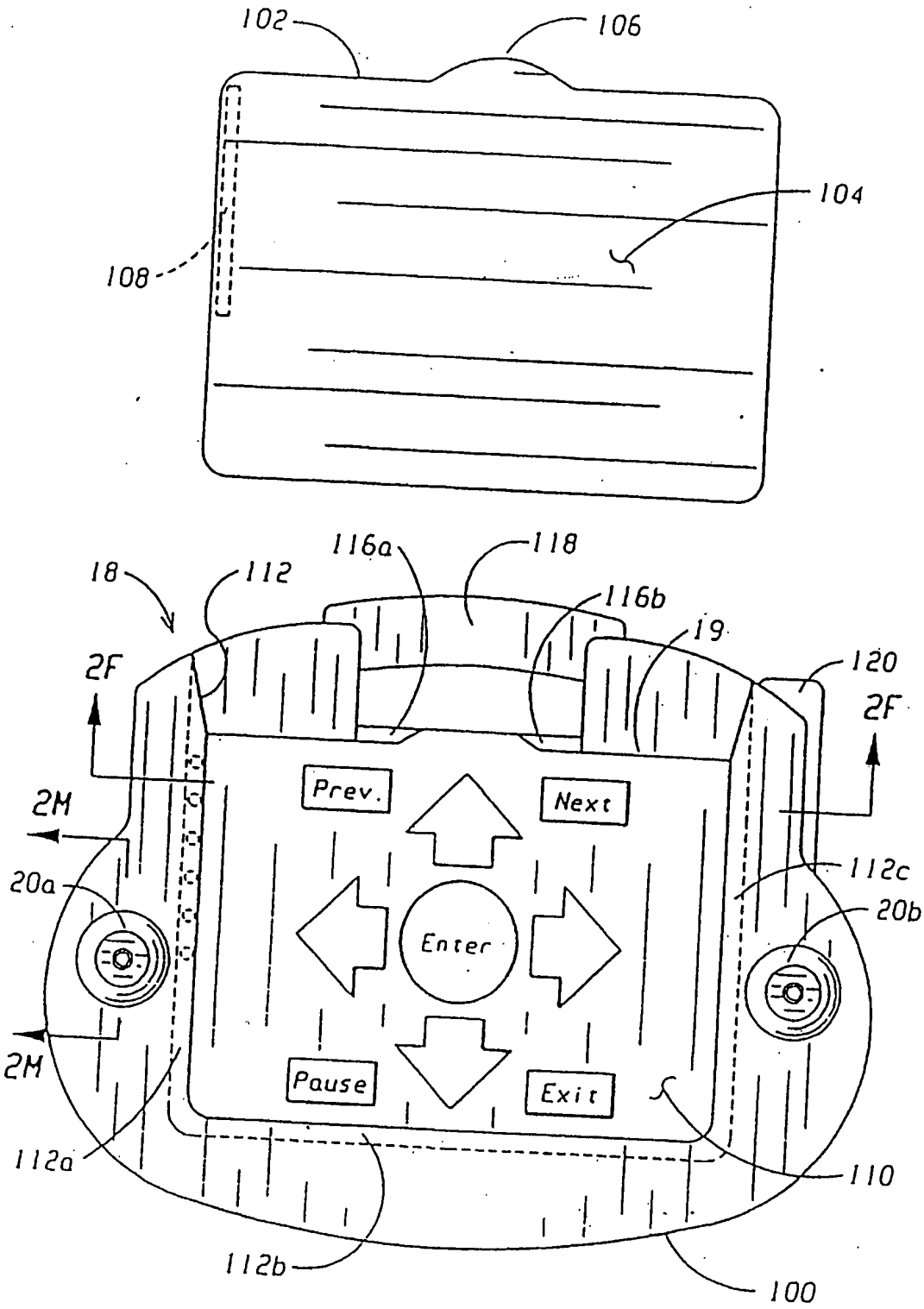

102
106
104
108
18
112
116a
118
116b
19
120
2F
2F
2M
2M
Prev.
Next
Enter
Pause
Exit
20a
20b
112c
112a
112b
110
100

图 2A

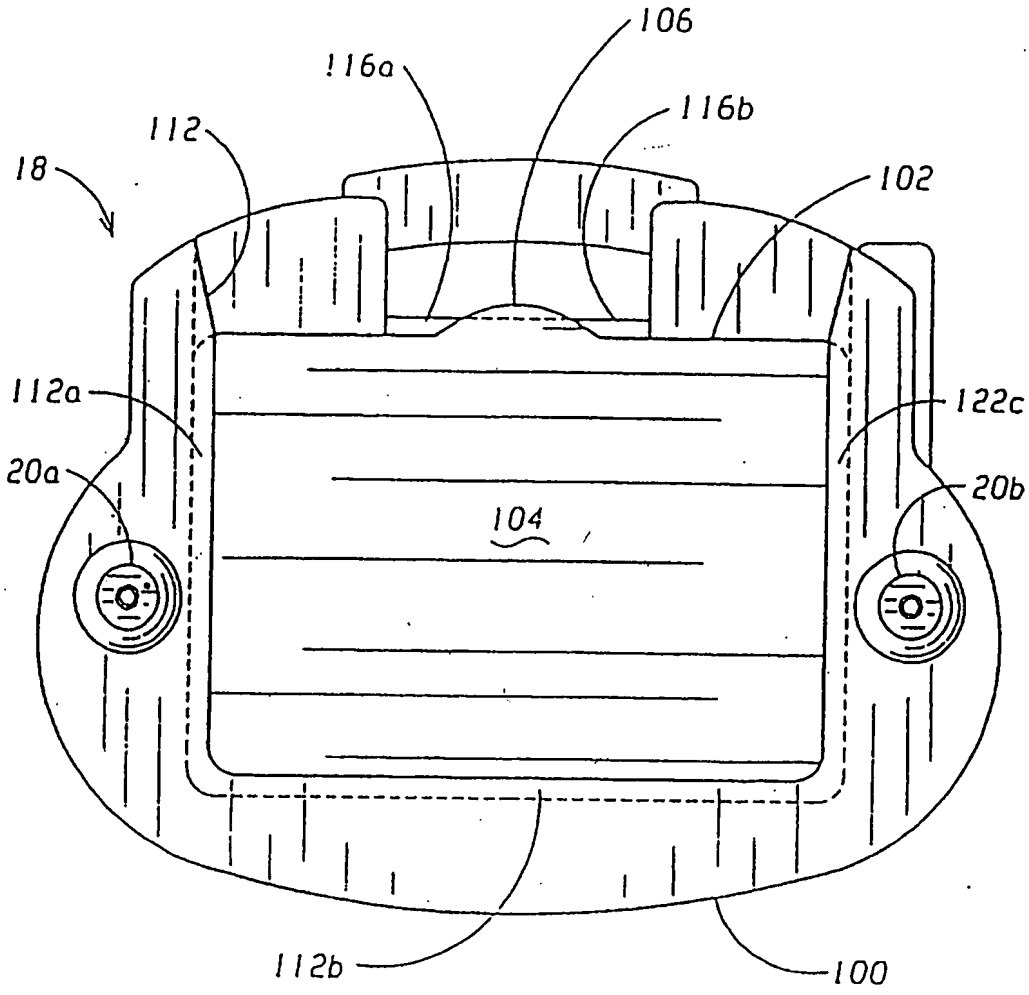


图 2B

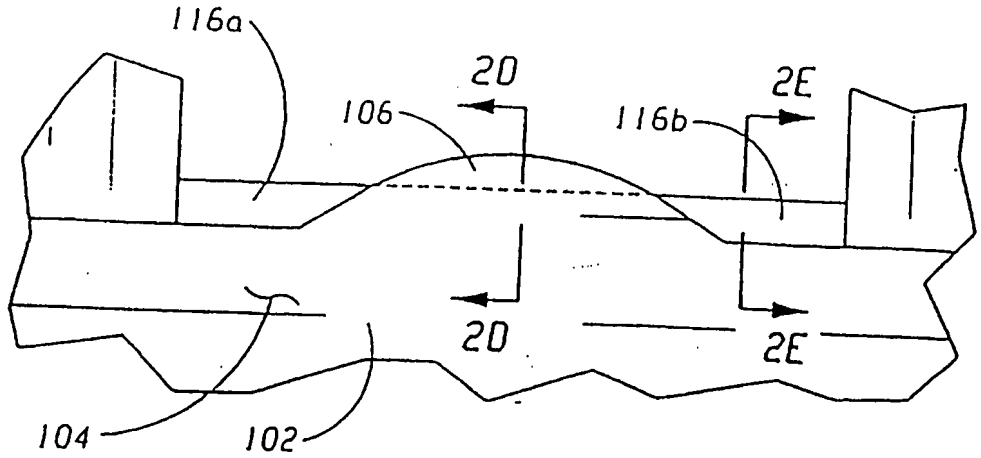


图 2C

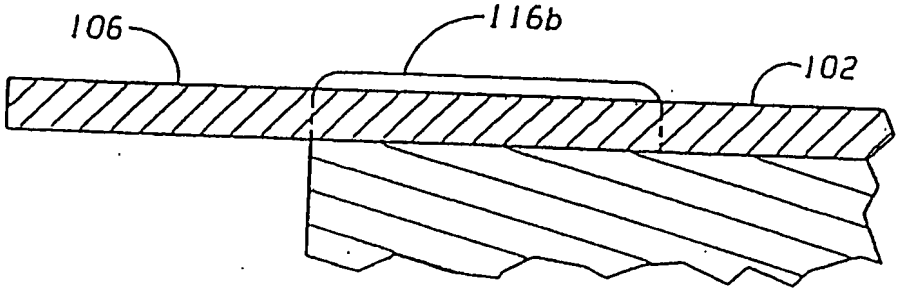


图 2D

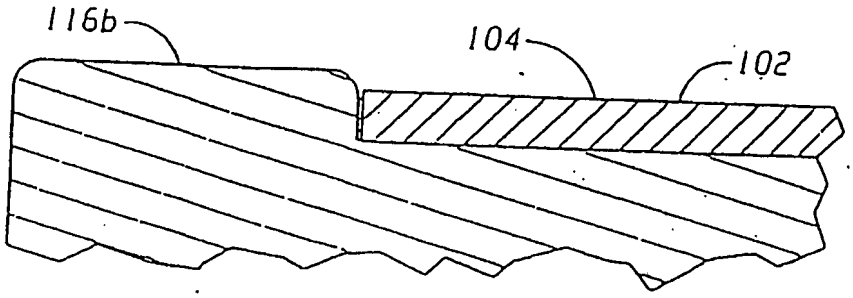


图 2E

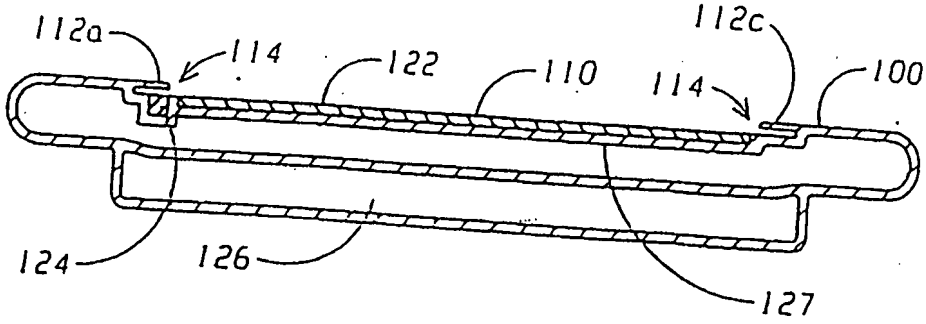

112a
114
122
110
114
112c
100
124
126
127

图 2F

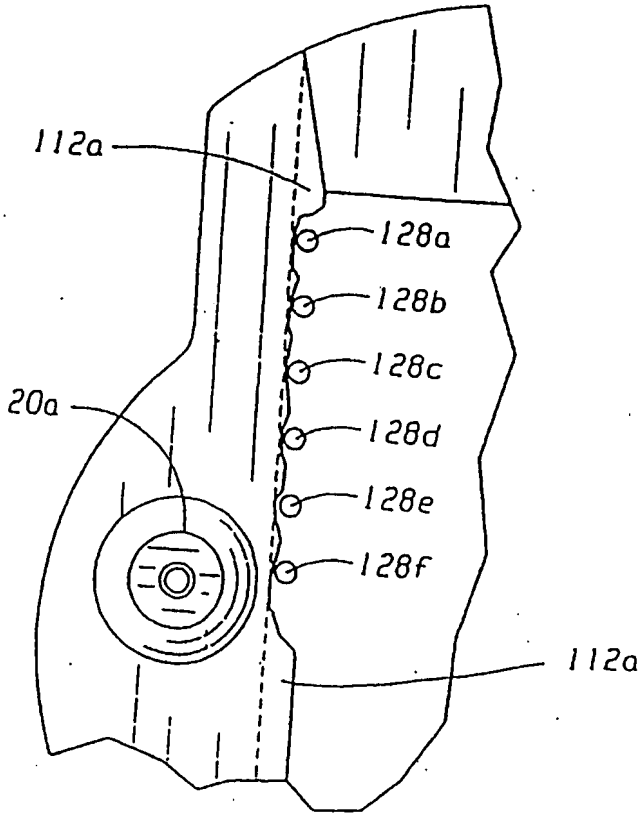

112a
128a
128b
128c
20a
128d
128e
128f
112a

图 2G

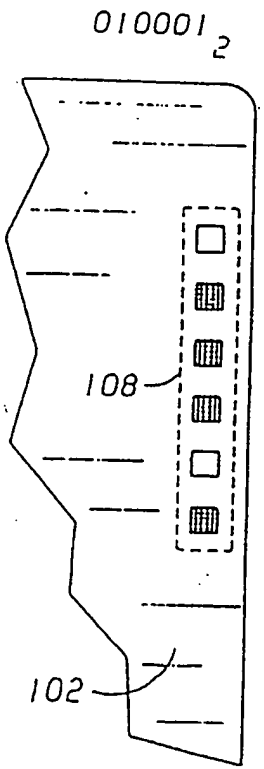


图 2H

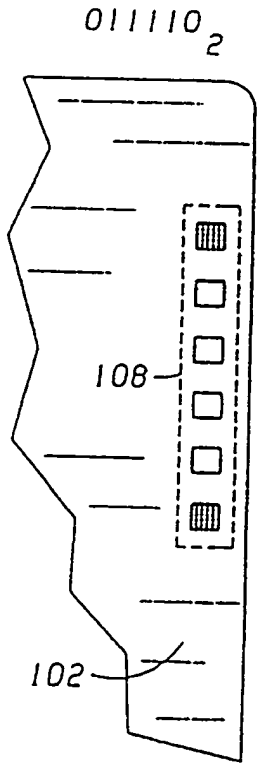


图 2I

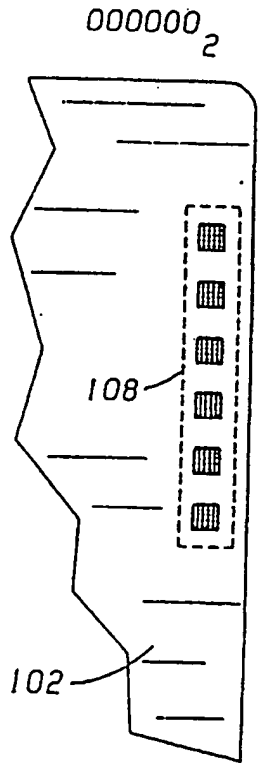


图 2J

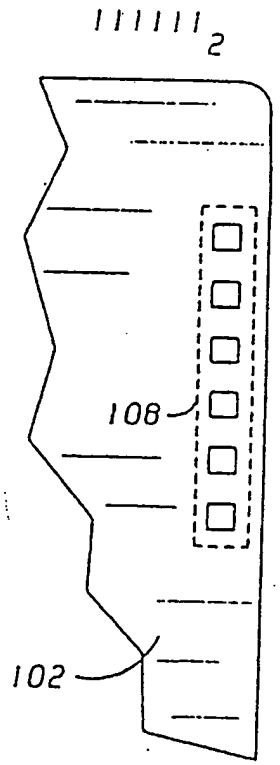


图 2K

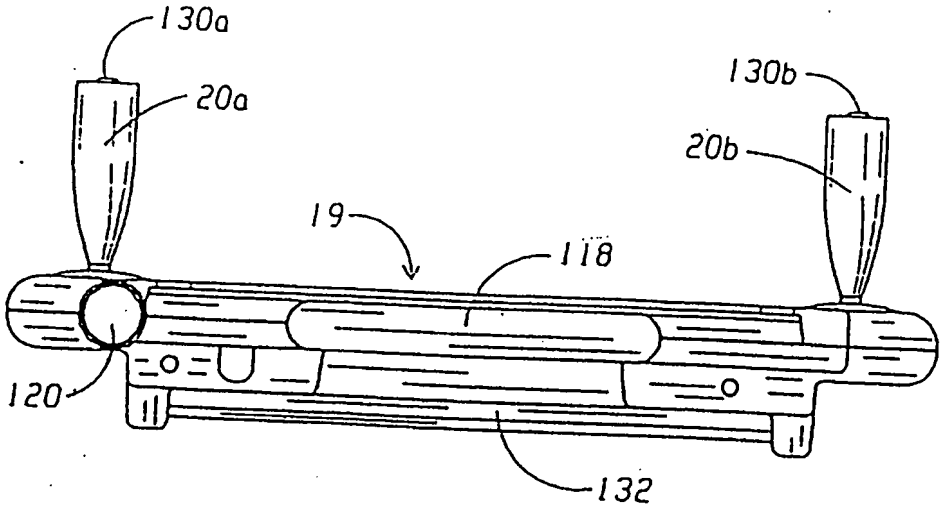

130a
20a
130b
20b
19
118
120
132

图 2L

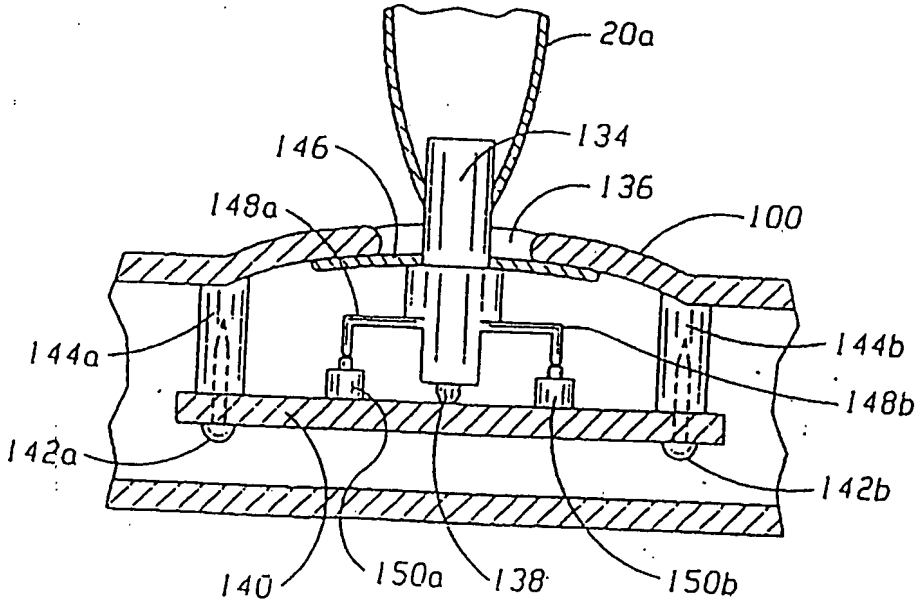

20a
134
146
136
148a
100
144a
144b
148b
142a
142b
140
150a
138
150b

图 2M

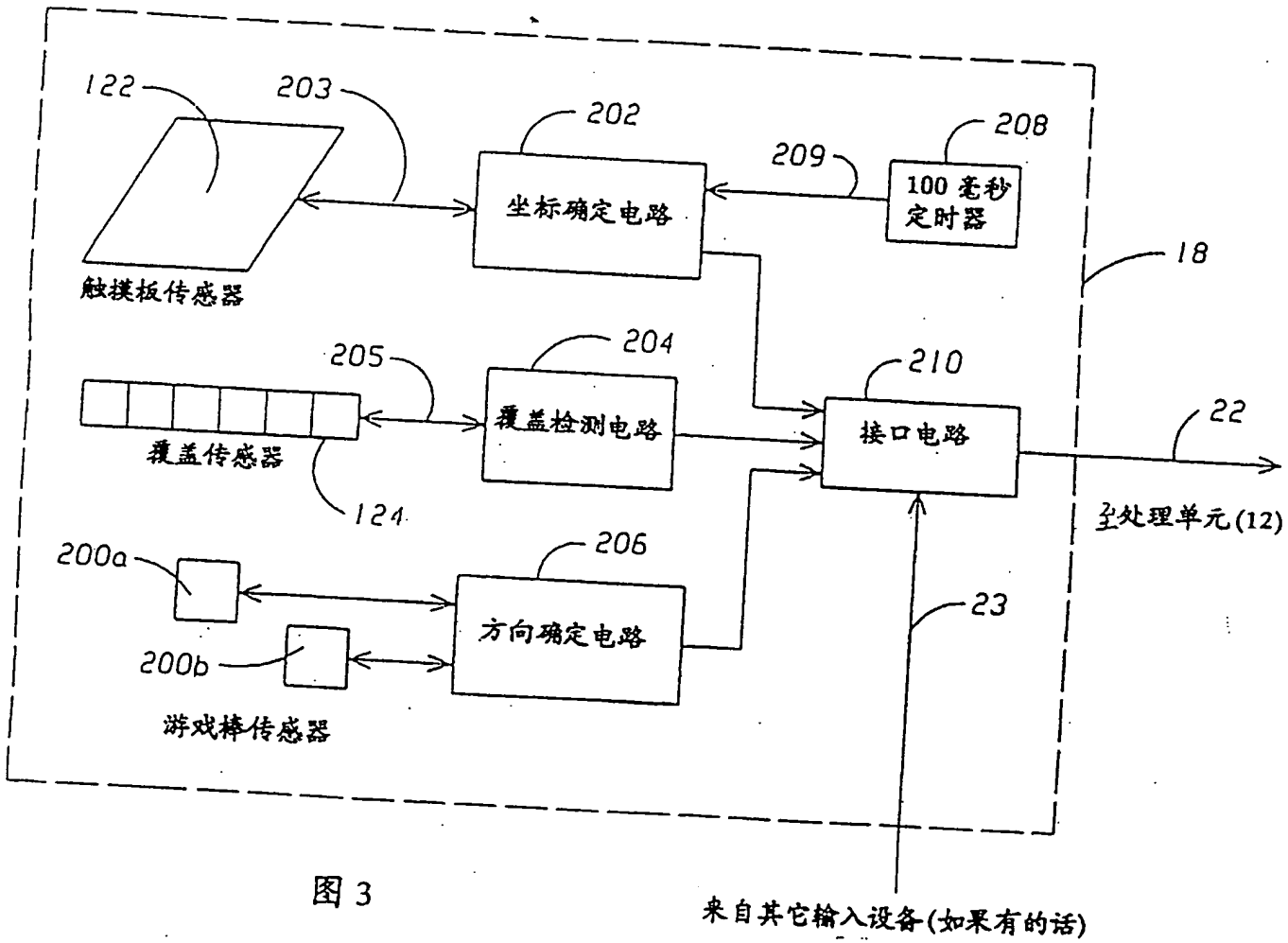

122
203
202
208
209
坐标确定电路
100 毫秒
定时器
触摸板传感器
18
205
204
210
覆盖检测电路
接口电路
22
覆盖传感器
124
至处理单元(12)
200a
206
方向确定电路
23
200b
游戏棒传感器
来自其它输入设备(如果有的话)

图 3

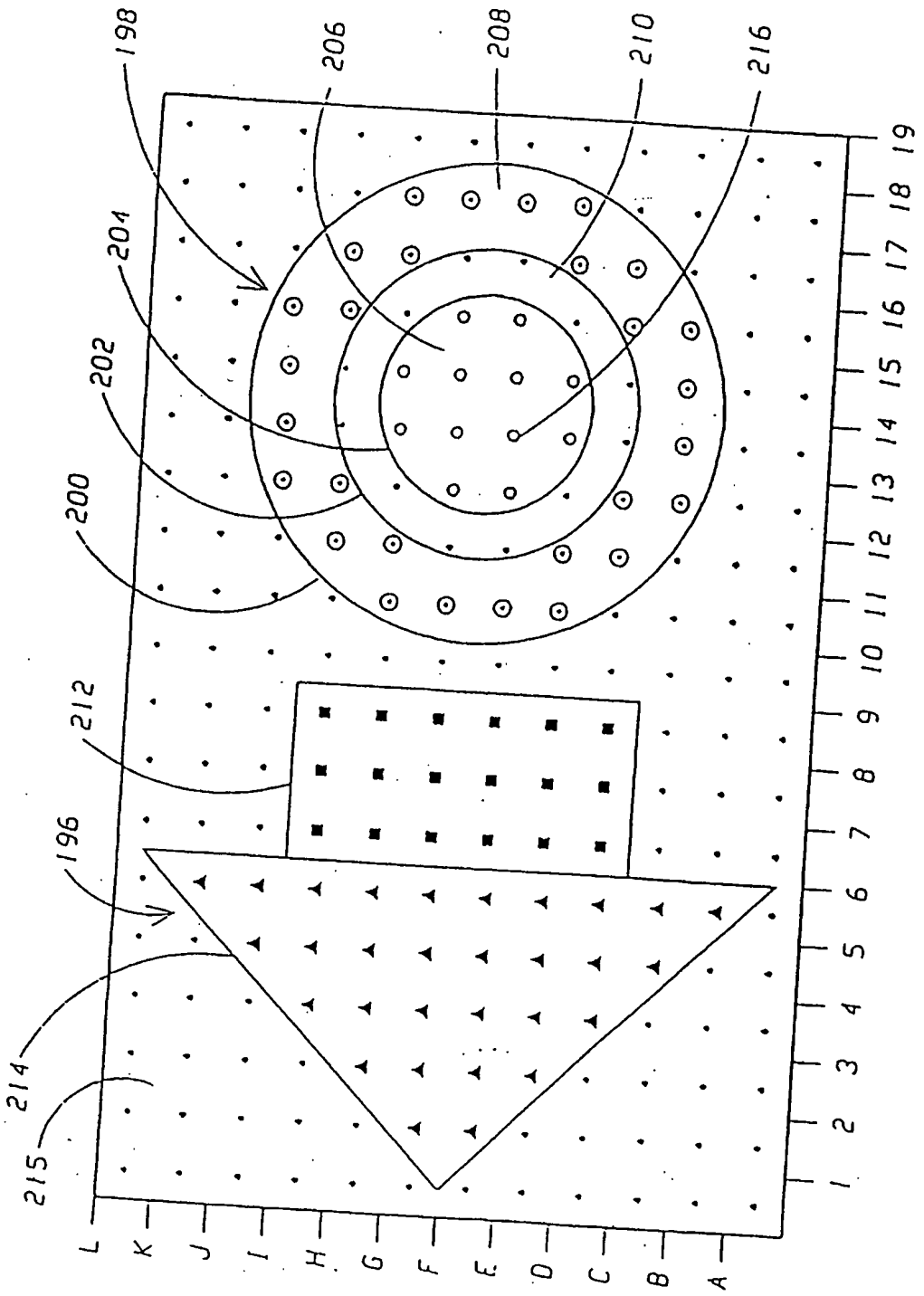

198
206
208
210
216
204
202
200
212
196
214
215
L
K
J
I
H
G
F
E
D
C
B
A
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

图 4